

REST APIs con Python (I)

Hay una cantidad increíble de datos disponibles en la web. Muchos servicios web, como YouTube y GitHub, hacen que sus datos sean accesibles a aplicaciones de terceros a través de una interfaz de programación de aplicaciones (API). Una de las formas más populares de construir APIs es con el estilo de arquitectura REST. Python proporciona algunas herramientas excelentes no sólo para obtener datos de las APIs REST, sino también para construir tus propias APIs REST en Python.

Utilizando Python y las APIs REST, puedes recuperar, analizar, actualizar y manipular los datos proporcionados por cualquier servicio web que te interese.

Arquitectura REST y servicios Web REST

Sólo para que lo tengas presente por si alguna vez lo quieres estudiar más a fondo aquí te dejo lo que se entiende por arquitectura REST, siendo las APIs REST aquellas que se han construido siguiendo esa filosofía. Nuestra aproximación va a ser más práctica, en las siguientes secciones vamos a ver qué necesitas para llamar a una API, cómo se hace y vamos a hacer un par de ejemplos con una API de juguete. En la siguiente sesión te presentaré un listado de APIs interesantes y jugaremos con un par de ellas. En los ejercicios verás más posibilidades y en la sesión en vivo hablaremos de las APIs que se invocan/utilizan de una forma diferente a la aquí presentada.

REST significa [representational state transfer](#) y es un estilo de arquitectura de software que define un patrón para las comunicaciones entre cliente y servidor a través de una red. REST proporciona un conjunto de restricciones para que la arquitectura del software promueva el rendimiento, la escalabilidad, la simplicidad y la fiabilidad del sistema.

Un servicio web REST es cualquier servicio web que se adhiere a las restricciones de la arquitectura REST. Estos servicios web exponen sus datos al mundo exterior a través de una API. Las APIs REST proporcionan acceso a los datos del servicio web a través de URLs web públicas.

Por ejemplo, esta es una de las URL de la API REST de GitHub:

```
https://api.github.com/users/<nombredeusuario>
```

Esta URL te permite acceder a información sobre un usuario específico de GitHub. El acceso a los datos de una API REST se realiza enviando una solicitud HTTP a una URL específica y procesando la respuesta.

Qué se necesita para usar una API (REST)

En general, necesitamos conocer:

- El endpoint o lo que es lo mismo la dirección tipo página web a la que dirigir nuestras peticiones
- El método que se requiere para hacer según qué cosas (esto lo vimos hace un par de sesiones: GET, POST, PUT, DELETE aunque hay más)
- Los argumentos o parámetros que requiera la API para funcionar (por ejemplo, el rango de fechas sobre los que queremos los datos)
- Normalmente una API-Key que es un código/password que se requiere para poder acceder a la API.

Veamos un poco de cada uno:

Endpoints

Como se ha comentado son esas direcciones de acceso. Ojo a veces estas direcciones (o URLs) tienen una pinta un poco más complicada que una dirección normal. Pongamos como ejemplo la API Countries, esta API proporciona información detallada sobre los países del mundo. Es una excelente opción para obtener datos geográficos, políticos y económicos de diferentes naciones.

Esta API tiene una dirección base:

- **URL Base:** <https://restcountries.com/v3.1/>

Y a partir de ella se construyen los diferentes endpoints en el que cada uno ofrece información diferente. Ahora sólo quédate con la parte de endpoint (que se construye concatenando la URL base anterior al campo endpoint que aparece más abajo)

Endpoints Principales de restcountries:

1. Obtener todos los países:

- **Endpoint:** /all
- **Método:** GET
- **Uso:** Devuelve información sobre todos los países.
- **Ejemplo:** <https://restcountries.com/v3.1/all>

2. Buscar país por nombre:

- **Endpoint:** /name/{name}
- **Método:** GET
- **Uso:** Busca países por su nombre.
- **Ejemplo:** <https://restcountries.com/v3.1/name/norway> (para buscar información sobre Noruega).

3. Buscar países por moneda:

- **Endpoint:** /currency/{currency}
- **Método:** GET
- **Uso:** Busca países que utilizan una moneda específica.
- **Ejemplo:** <https://restcountries.com/v3.1/currency/euro> (para buscar países que usan el euro).

4. Buscar países por idioma:

- **Endpoint:** /lang/{language}
- **Método:** GET
- **Uso:** Busca países donde se habla un idioma específico.
- **Ejemplo:** <https://restcountries.com/v3.1/lang/spanish> (para buscar países donde se habla español).

5. Buscar países por capital:

- **Endpoint:** /capital/{capital}
- **Método:** GET

- **Uso:** Busca países por su capital.
- **Ejemplo:** <https://restcountries.com/v3.1/capital/oslo> (para buscar información sobre el país cuya capital es Oslo).

Métodos

Si te fijas en los endpoints anteriores todos requieren el método GET porque todos son de obtención de datos. Habrá APIs que permitan PUSH, PUT y DELETE para poder enviar datos o borrarlos, pero en general y por ahora nosotros nos centramos en la obtención de información y usaremos APIs y endpoints con el método GET, salvo que se diga lo contrario. Cuando llegue el momento de construir nuestras APIs entraremos en el resto de los métodos básicos.

Por terminar de aclarar, en la llamada a la API (en breve lo veremos, tendremos que especificar un método, que generalmente será GET)

Parámetros

Son los valores adicionales que hay que proporcionar para obtener resultados, siguiendo con el ejemplo anterior, ojo y para esa API en concreto, son parámetros todos los elementos que van entre llaves: name, country, currency,...

Hay varias formas de pasar los parámetros y en general depende de la API, pero se puede:

- Como parte de la URL: hacer como en el ejemplo (completando la dirección)
- Como una querystring dentro la URL: usando las sintaxis `endpoint?<nombre_parámetro>=<valor_parámetro>` (esto permite dar varios valores si hay más de un parámetro, por ejemplo: `endpoint?<nombre_parametro_1>=<valor_parametro_1>&?<nombre_parametro_2>=<valor_parametro2>`) (si, ahora cuando te fijas en una dirección web podrás ver que le estás pasando parámetros después de los "?=")
- Como parte del cuerpo de la petición: usando un diccionario con la sintaxis: `{"<nombre_parametro_1>": valor_parametro_1,...}`

[Nota en el caso de Rest Countries sólo es válido el primero]

En los ejemplos te quedará más claro.

API-Key/API-Token

Rest Countries no necesita API-Key pero existen muchos otros servicios o APIs que si lo requieren. Un servicio que requiere una API key y es gratuita y fácil de obtener (pero tienes que pasar un proceso de registro) es la API de The Movie Database (TMDb). Esta API proporciona información extensa sobre películas, programas de televisión, actores, y mucho más.

Para registrarte puedes dirigirte a: <https://developer.themoviedb.org/docs>

Después de seguir los pasos, aceptar las políticas de uso y demás te ofrecerán dos cosas: una API-Key y un API-Token, la primera se incluye como parámetro adicional en cada llamada a la API, el segundo se coloca en el

de la petición. Son dos formas independientes de autenticarse y poder usar la API. Lo veremos en los ejercicios extra, para poder hacerlos deberás tener tu API-Key o tu API-token.

Algunos endpoints de TMDb (no confundir con IMDB):

1. Obtener Detalles de una Película:

- **Endpoint:** /movie/{movie_id}
- **Método:** GET
- **Uso:** Obtiene detalles específicos de una película utilizando su ID.
- **Ejemplo:** https://api.themoviedb.org/3/movie/550?api_key=tu_api_key

2. Buscar Películas:

- **Endpoint:** /search/movie
- **Método:** GET
- **Uso:** Busca películas por título.
- **Ejemplo:** https://api.themoviedb.org/3/search/movie?query=Inception&api_key=tu_api_key

Cómo se usa una API (REST): Rest Countries

Para invocar una API usamos request de la librería requests (ojo a la s final). Vamos a ver un ejemplo muy sencillo de cómo usar esta API y en la siguiente píldora haremos varias consultas diferentes para que veas más formas de acceder:

```
import requests
base_url="https://restcountries.com/v3.1/"
endpoint="name/"
parametro="Japan"
url=base_url+endpoint+parametro
print(url)
response_api=requests.request("GET",url)
print(response_api)
```

<https://restcountries.com/v3.1/name/Japan>
<Response [200]>

De las sesiones anteriores, recordarás que cada respuesta tiene su `status_code`, su `text` y su método `json`. Veamos que nos ha devuelto para cada invocación:

```
print(response_api.text)
print("\n"*3)
print(response_api.json())
```

```
[{"name":{"common":"Japan","official":"Japan","nativeName":{"jpn":{"official":"日本","common":"日本"}}},"tld":[".jp",".みんな"],"cca2":"JP","ccn3":"392","cca3":"JPN","cioc":"JPN","independent":true,"status":"officially-assigned","unMember":true,"currencies":{"JPY":{"name":"Japanese yen","symbol":"¥"},"id":{"root":"+8","suffixes":["1"]},"capital":["Tokyo"],"altSpellings":["JP","Nippon","Nihon"],"region":"Asia","subregion":"Eastern Asia","languages":{"jpn":"Japanese"},"translations":{"ara":{"official":"اليابان","common":"اليابان"},"bre":{"official":"Japan","common":"Japan"},"ces":{"official":"Japonsko","common":"Japonsko"},"cym":{"official":"Japan","common":"Japan"},"deu":{"official":"Japan","common":"Japan"},"est":{"official":"Japan","common":"Jaapan"},"fin":{"official":"Japani","common":"Japani"},"fra":{"official":"Japon","common":"Japon"},"ger":{"official":"Japan","common":"Japan"},"grk":{"official":"Ιαπωνία","common":"Ιαπωνία"},"heb":{"official":"יפן","common":"יפן"},"hrv":{"official":"Japan","common":"Japan"},"ice":{"official":"Japan","common":"Japan"},"ita":{"official":"Giappone","common":"Giappone"},"jpn":{"official":"日本","common":"日本"},"kor":{"official":"일본","common":"일본"},"nld":{"official":"Japan","common":"Japan"},"nor":{"official":"Japan","common":"Japan"},"per":{"official":"ژاپن","common":"ژاپن"},"pol":{"official":"Japon","common":"Japon"},"por":{"official":"Japão","common":"Japão"},"rus":{"official":"Япония","common":"Япония"},"slk":{"official":"Japonsko","common":"Japonsko"},"spa":{"official":"Japón","common":"Japón"},"swe":{"official":"Japan","common":"Japan"},"tur":{"official":"Japonya","common":"Japonya"},"ukr":{"official":"Японія","common":"Японія"},"uzb":{"official":"Япония","common":"Япония"},"vie":{"official":"Nhật Bản","common":"Nhật Bản"},"zsm":{"official":"Jepun","common":"Jepun"}}},"root":"+8","suffixes":["1"]},"capital":["Tokyo"],"altSpellings":["JP","Nippon","Nihon"],"region":"Asia","subregion":"Eastern Asia","languages":{"jpn":"Japanese"},"translations":{"ara":{"official":"اليابان","common":"اليابان"},"bre":{"official":"Japan","common":"Japan"},"ces":{"official":"Japonsko","common":"Japonsko"},"cym":{"official":"Japan","common":"Japan"},"deu":{"official":"Japan","common":"Japan"},"est":{"official":"Japan","common":"Jaapan"},"fin":{"official":"Japani","common":"Japani"},"fra":{"official":"Japon","common":"Japon"},"ger":{"official":"Japan","common":"Japan"},"grk":{"official":"Ιαπωνία","common":"Ιαπωνία"},"heb":{"official":"יפן","common":"יפן"},"hrv":{"official":"Japan","common":"Japan"},"ice":{"official":"Japan","common":"Japan"},"ita":{"official":"Giappone","common":"Giappone"},"jpn":{"official":"日本","common":"日本"},"kor":{"official":"일본","common":"일본"},"nld":{"official":"Japan","common":"Japan"},"nor":{"official":"Japan","common":"Japan"},"per":{"official":"ژاپن","common":"ژاپن"},"pol":{"official":"Japon","common":"Japon"},"por":{"official":"Japão","common":"Japão"},"rus":{"official":"Япония","common":"Япония"},"slk":{"official":"Japonsko","common":"Japonsko"},"spa":{"official":"Japón","common":"Japón"},"swe":{"official":"Japan","common":"Japan"},"tur":{"official":"Japonya","common":"Japonya"},"ukr":{"official":"Японія","common":"Японія"},"uzb":{"official":"Япония","common":"Япония"},"vie":{"official":"Nhật Bản","common":"Nhật Bản"},"zsm":{"official":"Jepun","common":"Jepun"}}}]
```

```
n": "Japan"}, {"hrv": {"official": "Japan", "common": "Japan"}, {"hun": {"official": "Japán", "common": "Japán"}, {"ita": {"official": "Giappone", "common": "Giappone"}, {"jpn": {"official": "日本", "common": "日本"}, {"kor": {"official": "일본국", "common": "일본국"}, {"nld": {"official": "Japan", "common": "Japan"}, {"per": {"official": "ژاپن", "common": "ژاپن"}, {"pol": {"official": "Japonia", "common": "Japonia"}, {"por": {"official": "Japão", "common": "Japão"}, {"rus": {"official": "Япония", "common": "Япония"}, {"slk": {"official": "Japonsko", "common": "Japonsko"}, {"spa": {"official": "Japón", "common": "Japón"}, {"srp": {"official": "Јапан", "common": "Јапан"}, {"swe": {"official": "Japan", "common": "Japan"}, {"tur": {"official": "Japonya", "common": "Japonya"}, {"urd": {"official": "جاپان", "common": "جاپان"}, {"zho": {"official": "日本国", "common": "日本"}}, {"latlng": [36.0, 138.0], "landlocked": false, "area": 377930.0, "demonyms": {"eng": {"f": "Japanese", "m": "Japanese"}, {"fra": {"f": "Japonaise", "m": "Japonais"}}, {"flag": "\uD83C\uDDFF\uD83C\uDDF5", "maps": {"googleMaps": "https://goo.gl/maps/NGTLCSSrA8bMrvnX9", "openStreetMaps": "https://www.openstreetmap.org/relation/382313"}, "population": 125836021, "gini": {"2013": 32.9}, {"fifa": "JPN", "car": {"signs": ["J"], "side": "left"}, "timezones": ["UTC+09:00"], "continents": ["Asia"], "flags": {"png": "https://flagcdn.com/w320/jp.png", "svg": "https://flagcdn.com/jp.svg", "alt": "The flag of Japan features a crimson-red circle at the center of a white field."}, {"coatOfArms": {"png": "https://mainfacts.com/media/images/coats_of_arms/jp.png", "svg": "https://mainfacts.com/media/images/coats_of_arms/jp.svg"}, {"startOfWeek": "monday", "capitalInfo": {"latlng": [35.68, 139.75]}, {"postalCode": {"format": "###-####", "regex": "^(\\d{7})$"}}}
```

```
{'name': {'common': 'Japan', 'official': 'Japan', 'nativeName': {'jpn': {'official': '日本', 'common': '日本'}}, 'tld': ['.jp', '.みんな'], 'cca2': 'JP', 'ccn3': '392', 'cca3': 'JPN', 'cioc': 'JPN', 'independent': True, 'status': 'officially-assigned', 'unMember': True, 'currencies': {'JPY': {'name': 'Japanese yen', 'symbol': '¥'}}, 'idd': {'root': '+8', 'suffixes': ['1']}, 'capital': ['Tokyo'], 'altSpellings': ['JP', 'Nippon', 'Nihon'], 'region': 'Asia', 'subregion': 'Eastern Asia', 'languages': {'jpn': 'Japanese'}, 'translations': {'ara': {'official': 'اليابان', 'common': 'اليابان'}, 'bre': {'official': 'Japan', 'common': 'Japan'}, 'ces': {'official': 'Japonsko', 'common': 'Japonsko'}, 'cym': {'official': 'Japan', 'common': 'Japan'}, 'deu': {'official': 'Japan', 'common': 'Japan'}, 'est': {'official': 'Jaapan', 'common': 'Jaapan'}, 'fin': {'official': 'Japani', 'common': 'Japani'}, 'fra': {'official': 'Japon', 'common': 'Japon'}, 'hrv': {'official': 'Japan', 'common': 'Japan'}, 'hun': {'official': 'Japán', 'common': 'Japán'}, 'ita': {'official': 'Giappone', 'common': 'Giappone'}, 'jpn': {'official': '日本', 'common': '日本'}, 'kor': {'official': '일본국', 'common': '일본'}, 'nld': {'official': 'Japan', 'common': 'Japan'}, 'per': {'official': 'ژاپن', 'common': 'ژاپن'}, 'pol': {'official': 'Japonia', 'common': 'Japonia'}, 'por': {'official': 'Japão', 'common': 'Japão'}, 'rus': {'official': 'Япония', 'common': 'Япония'}, 'slk': {'official': 'Japonsko', 'common': 'Japonsko'}, 'spa': {'official': 'Japón', 'common': 'Japón'}, 'srp': {'official': 'Јапан', 'common': 'Јапан'}, 'swe': {'official': 'Japan', 'common': 'Japan'}, 'tur': {'official': 'Japonya', 'common': 'Japonya'}, 'urd': {'official': 'جاپان', 'common': 'جاپان'}, 'zho': {'official': '日本国', 'common': '日本'}}, {"latlng": [36.0, 138.0], "landlocked": False, "area": 377930.0, "demonyms": {"eng": {"f": "Japanese", "m": "Japanese"}, {"fra": {"f": "Japonaise", "m": "Japonais"}}, {"flag": 'JP', "maps": {"googleMaps": "https://goo.gl/maps/NGTLCSSrA8bMrvnX9", "openStreetMaps": "https://www.openstreetmap.org/relation/382313"}, {"population": 125836021, "gini": {"2013": 32.9}, {"fifa": "JPN", "car": {"signs": ["J"], "side": "left"}, {"timezones": ["UTC+09:00"], "continents": ["Asia"], "flags": {"png": "https://flagcdn.com/w320/jp.png", "svg": "https://flagcdn.com/jp.svg", "alt": "The flag of Japan features a crimson-red circle at the center of a white field."}, {"coatOfArms": {"png": "https://mainfacts.com/media/images/coats_of_arms/jp.png", "svg": "https://mainfacts.com/media/images/coats_of_arms/jp.svg"}, {"startOfWeek": 'monday', "capitalInfo": {"latlng": [35.68, 139.75]}, {"postalCode": {"format": '###-####', "regex": "^(\\d{7})$"}}}
```

```
print(type(response_api.text))
<class 'str'>
print(type(response_api.json()))
<class 'list'>
```

En este caso el atributo text y la invocación del método json nos devuelven lo mismo. Ahora ya podríamos obtener información de los json, como aprendimos al tratar Json (básicamente son diccionarios) y ver la forma de pasar los datos a un DataFrame. En general no iremos a ciegas, sino que siempre que quieras usar una API consulta antes su documentación para saber qué formato tiene la salida de cada endpoint. (Para este caso puedes encontrar [aquí](#) una descripción de cada campo)

Cómo se encuentra una API y algunas APIs interesantes

La mayoría de servicios populares tiene una o varias APIs, busca en sus barras de búsqueda o en sus menús de la página principal bajo el epígrafe API o similar. Además de mostrarte como acceder (si hay que registrarse, etc) te dará la forma de llegar a la documentación (con los endpoints, métodos, resultados esperados, códigos de status y su significado, etc). Aquí te dejo una lista de algunas APIs "populares"

Claro, aquí tienes una lista de APIs populares junto con sus URL base. Estas APIs abarcan una variedad de usos, desde datos geográficos hasta entretenimiento y redes sociales:

1. **OpenWeatherMap (Datos Meteorológicos)**
 - URL Base: <http://api.openweathermap.org/data/2.5/>
 - [Sitio Web](#)
2. **GitHub API (Interacción con Repositorios y Usuarios de GitHub)**
 - URL Base: <https://api.github.com/>
 - [Sitio Web](#)
3. **Spotify Web API (Datos de Música y Playlists)**
 - URL Base: <https://api.spotify.com/v1/>
 - [Sitio Web](#)
4. **Twitter API (Acceso a Datos de Twitter)**
 - URL Base: <https://api.twitter.com/>
 - [Sitio Web](#)
5. **Google Maps API (Servicios de Mapas y Localización)**
 - URL Base: Varía según el servicio (por ejemplo, Maps, Routes, Places)
 - [Sitio Web](#)
6. **Unsplash API (Fotos de Alta Resolución Gratuitas)**
 - URL Base: <https://api.unsplash.com/>
 - [Sitio Web](#)
7. **NASA API (Imágenes y Datos de la NASA)**
 - URL Base: <https://api.nasa.gov/>
 - [Sitio Web](#)
8. **INE API (Instituto Nacional de Estadística de España)**
-[Sitio Web](#)
9. **Idealista**
-[Sitio Web](#)