

Ejemplo de uso de API rest

Para familiarizarnos con otras situaciones de uso de las APIs vamos a trabajar con la API de la AEMET. Será un ejemplo sencillo y ya ampliaremos el uso de APIs en los ejercicios del workout y en la práctica obligatoria.

```
import pandas as pd
import requests
```

AEMET

La AEMET tiene unos cuantos endpoints con previsiones del tiempo en municipios, zona de montaña, playas, etc. [Aquí](#) puedes encontrar el listado de los mismos con información adicional. Para este ejemplo vamos a emplear el endpoint que da la predicción del tiempo en las playas:

```
base_url = "https://opendata.aemet.es/opendata"
endpoint_playa = "/api/prediccion/especifica/playa/{playa}" # La predicción diaria de la playa que se
pasa como parámetro
```

Si consultas la información verás que el parámetro playa es un identificador específico de AEMET de las playas y que se encuentra en:

["https://www.aemet.es/documentos/es/eltiempo/prediccion/playas/Playas_codigos.csv"](https://www.aemet.es/documentos/es/eltiempo/prediccion/playas/Playas_codigos.csv)

Es decir, necesitamos hacernos con esos códigos para poder consultar la API correctamente. Lo cual ya nos plantea una de las situaciones que nos encontraremos más de una vez, que las APIs a su vez nos van a hacer tener que buscar otras fuentes con códigos especiales (sobre todo las de fuentes públicas institucionales y gubernamentales que manejan codificaciones propias)

Además, a esa situación hay que añadir que para poder consultar la API de la Agencia Meteorológica necesitamos una API-Key, que si has hecho los deberes ya tendrás. Vayamos con esto primero y luego con el tema de los códigos. Sustituye tu API-Key en la siguiente celda y ejecútala:

```
tu_api_key =
"eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhbGJhLmJhcnJlcmFzQGdtYWlsLmNvbSIsImp0aSI6IjE1Y2Q3NjRkLWlxOWEtNDdmMi1iZTYzLWM2Yzc5YTk3OGZiOSIsImZcyiI6IjFFFTUVUliwiaWF0IjoxNzAwOTkwNzQ3LCJ1c2VySWQiOiIxNWNkNzY0ZC1iMTIhLTQ3ZjltYmU2My1jNmM3OWE5NzhmYjkiLCJyb2xlIjoiln0.L-W4CQWYRteFLKgfDTBXbJB5vuHYEquhrgW8gFZKn00"
```

Revisando la documentación de la API que ofrece la propia AEMET, cosa que tendremos que hacer con cualquier API que queramos usar, nos encontramos con un ejemplo de cómo usarla empleando la API-key:

```
url="https://opendata.aemet.es/opendata/api/valores/climatologicos/inventarioestacione
s/todasestaciones/"
```

```
querystring = {"api_key": "AQUI LA APIKEY"}
```

```
headers = {
```

```
'cache-control': "no-cache"
```

```
}
```

```
response = requests.request("GET", url, headers=headers, params=querystring)
```

Fíjate donde va la API-Key, que aparece un parámetro (el headers que es donde irá el API-Token si lo usásemos). Lo tenemos todo salvo el código de las playas... Y aquí es donde quiero enseñarte una capacidad adicional de los métodos read de Pandas:

```
url_codigos="https://www.aemet.es/documentos/es/eltiempo/prediccion/playas/Playas_codigos.csv"
```

```
df_codigos=pd.read_csv(url_codigos, encoding="latin1",sep=";")
```

```
df_codigos
```

```
[4]:
```

	ID_PLAYA	NOMBRE_PLAYA	ID_PROVINCIA	NOMBRE_PROVINCIA	ID_MUNICIPIO	NOMBRE_MUNICIPIO	LATITUD	LONGITUD
0	301101	Raco de l'Albir	3	Alacant/Alicante	3011	l'Alfàs del Pi	38° 34' 31"	-00° 03' 52"
1	301401	Sant Joan / San Juan	3	Alacant/Alicante	3014	Alicante/Alacant	38° 22' 48"	-00° 24' 32"
2	301408	El Postiguet	3	Alacant/Alicante	3014	Alicante/Alacant	38° 20' 46"	-00° 28' 38"
3	301410	Saladar	3	Alacant/Alicante	3014	Alicante/Alacant	38° 17' 02"	-00° 31' 08"
4	301808	La Roda	3	Alacant/Alicante	3018	Altea	38° 36' 29"	-00° 02' 16"
...
586	4807701	Plentzia-Plencia	48	Bizkaia	48077	Plentzia	43° 24' 49"	-02° 56' 45"
587	4808502	Atxabiribil-Arrietara	48	Bizkaia	48085	Sopela	43° 23' 24"	-02° 59' 43"
588	5100104	Benitez	51	Ceuta	51001	Ceuta	35° 53' 39"	-05° 20' 12"
589	5100108	La Ribera - El Chorrillo	51	Ceuta	51001	Ceuta	35° 53' 13"	-05° 19' 12"
590	5200103	El Hipódromo	52	Melilla	52001	Melilla	35° 17' 04"	-02° 56' 19"

591 rows × 8 columns

Perfecto, ahora lo que buscamos es guardar en un dataframe la predicción de las playas de Melilla:

```
df_codigos[df_codigos.NOMBRE_PROVINCIA=="Melilla"]
```

```
[6]:
```

	ID_PLAYA	NOMBRE_PLAYA	ID_PROVINCIA	NOMBRE_PROVINCIA	ID_MUNICIPIO	NOMBRE_MUNICIPIO	LATITUD	LONGITUD
590	5200103	El Hipódromo	52	Melilla	52001	Melilla	35° 17' 04"	-02° 56' 19"

```
codigo_playa=df_codigos.loc[df_codigos.NOMBRE_PROVINCIA=="Melilla","ID_PLAYA"].to_list()[0]
```

```
codigo_playa
```

```
5200103
```

Pues ahí lo tenemos, ahora solo nos queda construir la url e invocar la API:

```
url=base_url+endpoint_playa.replace("{playa}",f"{codigo_playa}")
```

```

print(url)
https://opendata.aemet.es/opendata/api/prediccion/especifica/playa/5200103

querystring = {"api_key": tu_api_key}

headers = {
    'cache-control': "no-cache"
}

response = requests.request("GET", url, headers=headers, params=querystring)

print(response.status_code)
200

```

Código correcto, obtengamos la información a través del método json() y veamos si lo podemos leer directamente con Pandas

```

datos=response.json()
datos

{'descripcion': 'exito',
 'estado': 200,
 'datos': 'https://opendata.aemet.es/opendata/sh/9c5dce78',
 'metadatos': 'https://opendata.aemet.es/opendata/sh/8ca2e7e3'}

```

Pues va a ser que no, los datos y los metadatos son otras dos direcciones http. ¿Cómo hacemos? Usaremos request una vez más para traernos esos datos

```

informacion=requests.get(datos["datos"])
informacion_dict=informacion.json()
informacion_dict

[{'origen': {'productor': 'Agencia Estatal de Meteorología - AEMET. Gobierno de España',
 'web': 'http://www.aemet.es',
 'language': 'es',
 'copyright': '© AEMET. Autorizado el uso de la información y su reproducción citando a AEMET como autora de la misma.',
 'notaLegal': 'http://www.aemet.es/es/nota\_legal',
 'elaborado': '2023-12-01T07:50:24',
 'nombre': 'El Hipódromo',
 'localidad': '-13535',
 'prediccion': {'dia': [{'estadoCielo': {'value': "",
 'f1': 120,
 'descripcion1': 'muy nuboso',
 'f2': 120,
 'descripcion2': 'muy nuboso'},
 'viento': {'value': "",
 'f1': 210,
 'descripcion1': 'flojo',

```

```
'f2': 210,
'descripcion2': 'flojo'},
'oleaje': {'value': '',
'f1': 310,
'descripcion1': 'débil',
'f2': 310,
'descripcion2': 'débil'},
'tMaxima': {'value': '', 'valor1': 25},
'sTermica': {'value': '',
'valor1': 460,
'descripcion1': 'calor agradable'},
'tAgua': {'value': '', 'valor1': 18},
'uvMax': {'value': '', 'valor1': 2},
'fecha': 20231201,
'tmaxima': {'value': '', 'valor1': 25},
'stermica': {'value': '',
'valor1': 460,
'descripcion1': 'calor agradable'},
'tagua': {'value': '', 'valor1': 18}},
{'estadoCielo': {'value': '',
'f1': 120,
'descripcion1': 'muy nuboso',
'f2': 100,
'descripcion2': 'despejado'},
'viento': {'value': '',
'f1': 220,
'descripcion1': 'moderado',
'f2': 220,
'descripcion2': 'moderado'},
'oleaje': {'value': '',
'f1': 310,
'descripcion1': 'débil',
'f2': 310,
'descripcion2': 'débil'},
'tMaxima': {'value': '', 'valor1': 19},
'sTermica': {'value': '', 'valor1': 440, 'descripcion1': 'fresco'},
'tAgua': {'value': '', 'valor1': 18},
'uvMax': {'value': '', 'valor1': 2},
'fecha': 20231202,
'tmaxima': {'value': '', 'valor1': 19},
'stermica': {'value': '', 'valor1': 440, 'descripcion1': 'fresco'},
'tagua': {'value': '', 'valor1': 18}},
{'estadoCielo': {'value': '',
'f1': 120,
'descripcion1': 'muy nuboso',
'f2': 110,
'descripcion2': 'nuboso'},
'viento': {'value': '',
'f1': 210,
'descripcion1': 'flojo',
```

```

'f2': 210,
'descripcion2': 'flojo'},
'oleaje': {'value': ",
'f1': 310,
'descripcion1': 'débil',
'f2': 310,
'descripcion2': 'débil'},
'tMaxima': {'value': ", 'valor1': 19},
'sTermica': {'value': ", 'valor1': 450, 'descripcion1': 'suave'},
'tAgua': {'value': ", 'valor1': 17},
'uvMax': {'value': ", 'valor1': 2},
'fecha': 20231203,
'tmaxima': {'value': ", 'valor1': 19},
'stermica': {'value': ", 'valor1': 450, 'descripcion1': 'suave'},
'tagua': {'value': ", 'valor1': 17}}}],
'id': 5200103}]

```

Nos quedamos el nombre de la playa:

```

nombre_playa=informacion_dict[0]["nombre"]
nombre_playa

```

'El Hipódromo'

Se trata de una estructura algo rebuscada, básicamente toda la información está en la clave "prediccion" y dentro de esta en "dia" y hay tres días. Saquemos la info por día de temperatura máxima, temperatura del Agua y sensacion térmica, que corresponden a las claves "tMaxima", "tAgua", "sTermica". Además, usaremos la aproximación lista de diccionarios:

```

lista_predicciones=[]
for datos_dia in informacion_dict[0]["prediccion"]["dia"]:
    datos_salida={
        "nombre": nombre_playa
    }
    datos_salida["Temperatura_max"]=datos_dia["tMaxima"]["valor1"]
    datos_salida["Fecha"]=datos_dia["fecha"]
    datos_salida["Sensacion_termica"]=datos_dia["sTermica"]["descripcion1"]
    datos_salida["Temperatura_agua"]=datos_dia["tAgua"]["valor1"]
    lista_predicciones.append(datos_salida.copy())

pd.DataFrame(lista_predicciones)

```

[22]:

	nombre	Temperatura_max	Fecha	Sensacion_termica	Temperatura_agua
0	El Hipódromo	25	20231201	calor agradable	18
1	El Hipódromo	19	20231202	fresco	18
2	El Hipódromo	19	20231203	suave	17

Y ahí lo tenemos.