

Guia prático de acesso e download de dados

Portal de Dados Abertos do RJ

Exemplo de conjunto: secc_doerj_i_lei

Objetivo	Mostrar como localizar, baixar e exibir dados em Python usando o conjunto secc_doerj_i_lei.
Formatos abordados	JSON e CSV.
Público-alvo	Usuários que desejam consumir dados do portal manualmente ou por script.

Observação: Neste conjunto específico, o portal publica os recursos em JSON. O CSV é obtido a partir da conversão do JSON no Python.

1. Visão geral do conjunto

O conjunto secc_doerj_i_lei reúne publicações do Diário Oficial do Estado do Rio de Janeiro, organizadas por ano. No portal, os recursos aparecem em arquivos anuais em formato JSON.

A forma mais comum de uso é:

- 1) abrir a página do conjunto;
- 2) identificar o recurso desejado;
- 3) copiar a URL direta do arquivo;
- 4) baixar e tratar o conteúdo com Python.

2. Formas de acesso aos dados

Há duas formas principais de acesso:

- Download direto do recurso: quando você já tem a URL do arquivo JSON.
- API do CKAN: quando deseja consultar os metadados do conjunto, listar recursos e automatizar a descoberta das URLs disponíveis.

3. Estrutura do JSON

Ao carregar o arquivo JSON, a estrutura vem em formato aninhado. Neste exemplo, a chave principal é "I - Lei". Dentro dela, cada registro contém um identificador e, em geral, uma lista com a data da publicação e o texto da lei.

4. Exemplo 1 - Baixar o JSON diretamente

Use requests para baixar o arquivo e converter o conteúdo para objeto Python:

Código Python

```
import requests

url_json = "https://dadosabertos.rj.gov.br/dataset/0cafaf37-87ce-4ed7-ae48-67cb8de396ad/resource/0f3bc714-08fa-4448-b777-a01cd21728b6/download/i_lei_2024_parte_1.json"

response = requests.get(url_json, timeout=60)
response.raise_for_status()

dados = response.json()

print(type(dados))
print(dados.keys()) # Ex.: dict_keys(['I - Lei'])
```

5. Exemplo 2 - Inspeccionar alguns registros

Depois de carregar o JSON, você pode inspecionar os primeiros itens para entender a estrutura.

Código Python

```
import requests

url_json = "https://dadosabertos.rj.gov.br/dataset/0cafaf37-87ce-4ed7-ae48-67cb8de396ad/resource/0f3bc714-08fa-4448-b777-a01cd21728b6/download/i_lei_2024_parte_1.json"

dados = requests.get(url_json, timeout=60).json()
bloco = dados["I - Lei"]

print("Quantidade de registros:", len(bloco))

for i, (id_registro, conteudo) in enumerate(bloco.items()):
    print(f"\nRegistro {i+1}")
    print("ID:", id_registro)
    print("Data:", conteudo[0])
    print("Texto (início):", conteudo[1][:300])

    if i == 2:
        break
```

6. Exemplo 3 - Transformar o JSON em DataFrame

Como o JSON é aninhado, o mais prático é planificar os dados antes de analisar.

Código Python

```
import requests
import pandas as pd

url_json = "https://dadosabertos.rj.gov.br/dataset/0cafaf37-87ce-4ed7-ae48-67cb8de396ad/resource/0f3bc714-08fa-4448-b777-a01cd21728b6/download/i_lei_2024_parte_1.json"

dados = requests.get(url_json, timeout=60).json()
bloco = dados["I - Lei"]

linhas = []

for id_registro, conteudo in bloco.items():
    data_publicacao = conteudo[0] if len(conteudo) > 0 else None
    texto = conteudo[1] if len(conteudo) > 1 else None

    linhas.append({
        "id_registro": id_registro,
        "tipo": "I - Lei",
        "data_publicacao": data_publicacao,
        "texto": texto
    })
```

```
    })

df = pd.DataFrame(linhas)

print(df.head())
print(df.info())
```

7. Exemplo 4 - Salvar em CSV

Neste conjunto, o CSV é gerado a partir do DataFrame criado com base no JSON.

Código Python

```
import requests
import pandas as pd

url_json = "https://dadosabertos.rj.gov.br/dataset/0cafaf37-87ce-4ed7-ae48-67cb8de396ad/resource/0f3bc714-08fa-4448-b777-a01cd21728b6/download/i_lei_2024_parte_1.json"

dados = requests.get(url_json, timeout=60).json()
bloco = dados["I - Lei"]

linhas = []
for id_registro, conteudo in bloco.items():
    linhas.append({
        "id_registro": id_registro,
        "tipo": "I - Lei",
        "data_publicacao": conteudo[0] if len(conteudo) > 0 else None,
        "texto": conteudo[1] if len(conteudo) > 1 else None
    })

df = pd.DataFrame(linhas)

df.to_csv("i_lei_2024_parte_1.csv", index=False, encoding="utf-8-sig")
print("CSV salvo com sucesso.")
```

8. Exemplo 5 - Ler o CSV gerado

Código Python

```
import pandas as pd

df = pd.read_csv("i_lei_2024_parte_1.csv", encoding="utf-8-sig")

print(df.head())
print(df.loc[0, "texto"][:500])
```

9. Exemplo 6 - Consultar o dataset pela API CKAN

Esse método é útil para descobrir automaticamente quais recursos estão disponíveis.

Código Python

```
import requests
import pandas as pd

url_api = "https://dadosabertos.rj.gov.br/api/3/action/package_show?id=secc_doerj_i_lei"

response = requests.get(url_api, timeout=60)
response.raise_for_status()

dados_api = response.json()

print("Sucesso:", dados_api["success"])

resultado = dados_api["result"]
print("Nome do dataset:", resultado["name"])
print("Título:", resultado["title"])

recursos = resultado["resources"]

df_recursos = pd.DataFrame([
    {
        "id": r.get("id"),
        "nome": r.get("name"),
        "formato": r.get("format"),
        "url": r.get("url")
    }
    for r in recursos
])

print(df_recursos.head())
```

10. Exemplo 7 - Baixar automaticamente todos os JSONs do conjunto

Código Python

```
import requests
import os

url_api = "https://dadosabertos.rj.gov.br/api/3/action/package_show?id=secc_doerj_i_lei"

dados_api = requests.get(url_api, timeout=60).json()
recursos = dados_api["result"]["resources"]

os.makedirs("downloads_i_lei", exist_ok=True)

for recurso in recursos:
    formato = str(recurso.get("format", "")).upper()
    nome = recurso.get("name", "arquivo")
    url = recurso.get("url")

    if formato == "JSON" and url:
        print(f"Baixando: {nome}")
```

```

resposta = requests.get(url, timeout=120)
resposta.raise_for_status()

nome_arquivo = url.split("/")[-1]
caminho = os.path.join("downloads_i_lei", nome_arquivo)

with open(caminho, "wb") as f:
    f.write(resposta.content)

print("Downloads concluídos.")

```

11. Exemplo 8 - Criar um resumo da tabela

Código Python

```

import requests
import pandas as pd

url_json = "https://dadosabertos.rj.gov.br/dataset/0cafaf37-87ce-4ed7-ae48-67cb8de396ad/resource/0f3bc714-08fa-4448-b777-a01cd21728b6/download/i_lei_2024_parte_1.json"

dados = requests.get(url_json, timeout=60).json()
bloco = dados["I - Lei"]

linhas = []
for id_registro, conteudo in bloco.items():
    texto = conteudo[1] if len(conteudo) > 1 else ""
    linhas.append({
        "id_registro": id_registro,
        "data_publicacao": conteudo[0] if len(conteudo) > 0 else None,
        "resumo": texto[:200].replace("\r", " ").replace("\n", " ")
    })

df = pd.DataFrame(linhas)

print(df.head(10))

```

12. Boas práticas

- Verifique primeiro o formato disponível no recurso antes de escrever o script.
- Use timeout nas requisições para evitar travamentos.
- Converta o JSON em DataFrame quando precisar analisar, filtrar ou exportar.
- Salve o CSV com encoding='utf-8-sig' para facilitar abertura no Excel.
- Quando for automatizar vários anos, consulte a API CKAN para descobrir os recursos dinamicamente.

13. Resumo final

Para esse conjunto, o caminho mais direto é baixar o JSON do recurso, carregar com requests, transformar em DataFrame com pandas e, se necessário, exportar para CSV. Quando a intenção for automatizar o processo, a API CKAN facilita a descoberta dos recursos disponíveis.