

طلبات مكتبة requests

مثل `requests` الشائعة الـ `requests` مكتبة جمـيـع تـنـفـيـذـيـة يمكنـكـ بـاـسـتـخـدـامـ بـالـكـاـمـلـ لـإـسـتـبـدـالـ وإـثـبـاتـ الـوـكـالـاتـ، اـسـتـخـدـامـ الـمـلـفـاتـ، تـحـمـيـلـ، بـيـانـاتـ مـعـ الـعـمـلـ الـرـؤـوسـ، معـالـجـةـ الـطـلـبـاتـ، مـنـ مـخـتـلـفـةـ أـنـوـاعـ إـرـسـالـ وـمـارـشـ. شـاـمـلـاـ دـلـيـلـاـ لـدـيـكـ أـنـ لـضـمـانـ بـخـطـوـةـ، خـطـوـةـ الـوـظـائـفـ هـذـهـ مـنـ لـكـلـ أـمـثـلـةـ عـبـرـ سـأـقـوـدـكـ أـدـنـاهـ، الـاسـتـجـابـاتـ.

مكتبة تثبيت 1: الخطوة

مستودعك: في الأمر هذا بـتـشـغـيـلـ قـمـ مـكـتـبـةـ تـثـبـيـتـ مـنـ تـأـكـدـ الـبـدـءـ، قـبـلـ

```
pip install requests
```

الـأـمـثـلـةـ. فـيـ نـغـوصـ دـعـونـاـ الـآنـ،

طلبات إرسال

بسـيـطـ: `GET` و `POST` طـلـبـ إـرـسـالـ كـيـفـيـةـ إـلـيـكـ إـلـخـ. `requests` مـثـلـ طـرـقـ جـمـيـعـ دـعـمـتـ

طلب

```
import requests

#           GET
response = requests.get('https://api.example.com/data')

#
print("Status Code:", response.status_code)
print("Response Body:", response.text)
```

طلب

```
#           POST
response = requests.post('https://api.example.com/submit')

print("Status Code:", response.status_code)
print("Response Body:", response.text)
```

رۇوس إضافە

معلومە ئىلى كقايىمە بىرسالا قم الـ `headers`. البىانات أو الـ `Authorization`, `Content-Type`, `User-Agent` ئىلى كقايىمە بىرسالا قم الـ `headers`.

```
#  
headers = {  
    'Authorization': 'Bearer my_token',  
    'Content-Type': 'application/json',  
    'User-Agent': 'MyApp/1.0'  
}  
  
#      GET  
response = requests.get('https://api.example.com/data', headers=headers)  
  
print("Status Code:", response.status_code)  
print("Response Headers:", response.headers)  
print("Response Body:", response.text)
```

بىيانات ئىرسال

يىحدى دى `JSON` معلومە استخدىم، `Content-Type` `application/json` تىلىقىيەتلىكىدۇ.

```
#      JSON  
data = {  
    'key1': 'value1',  
    'key2': 'value2'  
}  
  
#      POST      JSON  
response = requests.post('https://api.example.com/submit', json=data, headers=headers)  
  
print("Status Code:", response.status_code)  
print("Response JSON:", response.json())
```

الملفات تحويل

الثانية وضع في الملفات افتح files. معلومة استخدم، خيار مثل الملفات لتحويل
للموضوع. بالإضافة إلى بيانات وضع من

بساطة ملف تحويل

```
#  
  
files = {  
    'file': open('myfile.txt', 'rb')  
}  
  
#      POST  
  
response = requests.post('https://api.example.com/upload', files=files)  
  
print("Status Code:", response.status_code)  
print("Response Body:", response.text)
```

#

```
files['file'].close()
```

تلقي: الملف إغلاق يضم من with بيان استخدام به الـ `files` الموصى `POST` طريقة النموذج بيانات مع ملف تحويل

```
#  
  
form_data = {  
    'description': 'My file upload'  
}  
  
#  
  
with open('myfile.txt', 'rb') as f:  
    files = {  
        'file': f  
    }  
  
    response = requests.post('https://api.example.com/upload', data=form_data, files=files)  
  
print("Status Code:", response.status_code)  
print("Response Body:", response.text)
```

الوکالات استخدام

قائمة. مع `proxies` معلمات استخدم، وكالة إعدادات مثل `http://myproxy:8080` وكالة عبر الطلبات توجيه لـ إعدادات

```
#  
proxies = {  
    'http': 'http://myproxy:8080',  
    'https': 'https://myproxy:8080'  
}  
  
#  
response = requests.get('https://api.example.com/data', proxies=proxies)  
  
print("Status Code:", response.status_code)  
print("Response Body:", response.text)
```

الاستجابات وإثبات معالجة

يمكنك والمعجنات. الرؤوس، `HTTP` ببيانات الحال، رموز مثل الاستجابة تفاصييل إلى سهلًا وصولاً `requests` مكتبة توفر `assert` مثل الاستجابات، لتأكيد `HTTP` في بيان استخدم.

استجابات تحلي

```
response = requests.get('https://api.example.com/data')  
  
#           JSON  
if response.status_code == 200:  
    data = response.json() # / Python  
    print("JSON Data:", data)  
else:  
    print("Error:", response.status_code)
```

الاستجابة تفاصييل إثبات

```
response = requests.get('https://api.example.com/data')  
  
#  
assert response.status_code == 200, f"Expected 200, got {response.status_code}"
```

```

# JSON

data = response.json()

assert 'key' in data, "Key not found in response"
assert data['key'] == 'expected_value', "Value does not match"

# 

assert 'Content-Type' in response.headers, "Content-Type header missing"
assert response.headers['Content-Type'] == 'application/json', "Unexpected Content-Type"

# 

cookies = response.cookies
assert 'session_id' in cookies, "Session ID cookie missing"

print("All assertions passed!")

```

أو الشبكية الأخطاء لالتقاط try-except كتلة في الطلبات أغلف الأخطاء معالجة:

```

try:
    response = requests.get('https://api.example.com/data')
    response.raise_for_status() # 4xx/5xx
    data = response.json()
    print("Data:", data)
except requests.exceptions.RequestException as e:
    print("Request failed:", e)

```

شامل مثال

الاستجابات: وإثبات ال威كالات، الملف، تحويل الرؤوس، بين يجمع شامل مثال هنا

```

import requests

# 

headers = {
    'Authorization': 'Bearer my_token'
}

# 

form_data = {

```

```
'description': 'My file upload'
}

#
proxies = {
    'http': 'http://myproxy:8080',
    'https': 'https://myproxy:8080'
}

#
try:
    with open('myfile.txt', 'rb') as f:
        files = {'file': f}
        response = requests.post(
            'https://api.example.com/upload',
            headers=headers,
            data=form_data,
            files=files,
            proxies=proxies
        )
        response.raise_for_status()      # HTTP

    #
    data = response.json()
    assert 'upload_id' in data, "Upload ID not found in response"
    print("Upload successful, ID:", data['upload_id'])

except requests.exceptions.RequestException as e:
    print("Request failed:", e)
```

مج مجموعات إستبدال الاستخدام ل إعادة قابلة وظائف

وظائف: في الامن طقية بتحفيز قم الاجموعات، أو المحفوظة طلبات لتمثيل

```
def upload_file(file_path, description, token, proxy=None):
    headers = {'Authorization': f'Bearer {token}'}
    data = {'description': description}
    proxies = {'http': proxy, 'https': proxy} if proxy else None
```

```

try:
    with open(file_path, 'rb') as f:
        files = {'file': f}
        response = requests.post(
            'https://api.example.com/upload',
            headers=headers,
            data=data,
            files=files,
            proxies=proxies
        )
        response.raise_for_status()
        data = response.json()
        assert 'upload_id' in data, "Upload ID not found"
        return data['upload_id']

except requests.exceptions.RequestException as e:
    print("Request failed:", e)
    return None

#
upload_id = upload_file('myfile.txt', 'My file upload', 'my_token', 'http://myproxy:8080')
if upload_id:
    print("Upload successful, ID:", upload_id)

```

إضافية ميزات

الطلبات. عبر الرؤوس أو المجنات على لاحفاظ() requests.Session() استخدم: جلسات
 الاتصال. لمنع الطلبات إلى ثوانٍ timeout=5 أضف: الانتظار أوقات
 مع بدها التحكم: التوجيه إعادة allow_redirects=True/False.

ملخص

المفات وتحمیل، بيارات الرؤوس، تضمین get, post، طلبات إرسال إلخ. requests مكتبة مع
 لإعادة قابلة وظائف في الكود تنظيم المجنات الرؤوس، الاتصالات وإثبات تحليلاً الوكلات استخدام
 الاستخدام

وال töpf! لـ برمجة مناسبًا، مرنًا برمجيًا بدأ لك يوفر هذا