

طلبات مكتبة requests

مثل `requests` المكتبة باستخدام بالكامل لإستبدال وإثبات الوكالات، استخدام الملفات، تحميل، `requests` بيانات مع العمل الرؤوس، معالجة الطلبات، من مختلف أنواع إرسال وممارسًا. شاملًا دليلًا لذيك أن لضمان بخطوة، خطوة الوظائف هذه من لكل أمثلة عبر سأكودك أدناه، الاستجابات.

مكتبة تثبيت: 1 الخطة

مستودعك: في الأمر هذا بتشغيل قم `requests` مكتبة تثبيت من تأكيد البدء، قبل

```
pip install requests
```

الأمثلة. في نغوص دعونا الآن،

طلبات إرسال

بسيط: `requests` و `requests` طلب إرسال كيفية إليك! إلخ، `requests`، `requests`، `requests`، `requests` طرق جميع `requests` مكتبة دعمت

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)
```

رؤوس إضافة

headers معلقة إلى لتقائمة بإرسالها قم الممتخصصة. ال بيانات أو المحتوى، أنواع للتصديقي، غالبًا الرؤوس تستخدم

```
#
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)
```

بيانات إرسال

يحدد هذا json معلقة استخدم، ، application/json إلى Content-Type لتلقائيًا طلب في مثل في اختياري مثل في طلب في بيانات لإرسال

```
# 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 تحميل

ملفات تحميل مثل الملفات ل تحميل 'rb' الثنائي وضع في الملفات افتح files. مع لمة استخدم، في ، خياري مثل الملفات ل تحميل لملف موضوع. ال إضافية البيانات وضمن

بسيط ملف تحميل

```
#
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()
```

تلقائيًا: الملف إغلاق يضمن with بيان استخدام به الموضوع الطريقة النمذجة بيانات مع ملف تحميل

```
#
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)
```

الوكالات استخدام

قائمة. مع proxies مغلقة استخدم، وكالات إعدادات مثل وكالات عبر الطلبات توجيه لإعادة

```
#
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 والمعالجة الرؤوس، بيانات الحالة، رموز مثل الاستجابة تفاصيل إلى سهلًا وصولًا requests مكتبة توفر
اختبار `assert` بيانه استخدام. `assert` في `assert` تأكيد `assert` في `assert` استخدام.

الاستجابات 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 تفاصيل 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 أغل ف الأخطاء مع الجة:

```

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 الوكالات، الملف، تحميل 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)

```

مجموعات استبدال الاستخدام لإعادة قابلية وظائف

وظائف: في المنطقية بتحفيز قم المجموعات، أو المحفوفة 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

- ال طلبات. عبر ال رؤوس أو ال معججات على ال لحفاظ `requests.Session()` استخدم: جلسات
- ال تعلق. لمنع ال طلبات إلى `ثواني` `timeout=5` أضف: ال انظار أوقات
- ال إعادة قابلية وظيف في ال كود تنظيم `ال معججات ال رؤوس`، `ال حالة`، ال استجابات وإثبات تحليل `ال وكالات` استخد ام ال استخد ام `allow_redirects=True/False` مع بها ال تحكم: ال توجيه إعادة

ملخص

□ ال ملفات وتحميل، `بيانات الرؤوس`، تضمين `إل.خ.` `get, post` `ال` طلبات إرسال `ال` يمكنك: `requests` مكتبة مع لإعادة قابلية وظيف في ال كود تنظيم `ال معججات ال رؤوس`، `ال حالة`، ال استجابات وإثبات تحليل `ال وكالات` استخد ام ال استخد ام

وال توظيف! لل برمجة من اسبًا، `ال` - مرنا برمجيا بديلا لك يوفر هذا