

طلبات مكتبة requests

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

مكتبة تثبيت: 1 الخطوة

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

```
pip install requests
```

الأمثلة. في نغوص دعونا الآن،

طلبات إرسال

بسيط: `requests` و `requests` طلب إرسال كيفية إليك إلخ.، `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 معلمة إلى لتقائمة بإرسالها قم الممتخصة. البيانات أو المحتوي، أنواع للصدقي، 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 معلمة استخدم، ، علامة في ، اختياري مثل طلب في بيانات لإرسال application/json إلى Content-Type لتلقي

```
# 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

الثنائي وضع في الملفات افتح files. معاملة استخدم، في في خيار مثل الملفات لتحميل لل موضوع. الإضافية 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()
```

تلقائي: الملف إغلاق يضمن with بيان استخدام به الموضوع الطريقة النموذج 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 مغلدة استخدم، `proxies` 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، `headers` بيانات الحالة، رموز مثل الاستجابة تفاصيل إلى سهل الوصول `requests` مكتبة توفر `assert` بيان استخدام `assert` لتأكيد `assert` في `assert` بيان استخدام.

الاستجابات تحليل

```
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 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 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 المنطقة بتحتفي زقم المجموعات، أو المحفظة 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

```

