



Python Requests Modülü

Web isteklerinizi python ile yapın

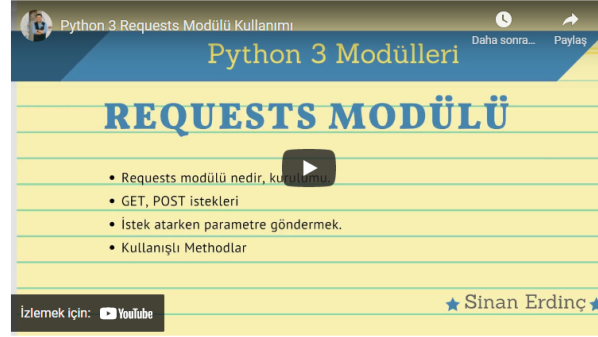
Tarih: 17-10-2017

Python, standart modüllerinin yanında harici yüzlerce kullanışlı modül ile birlikte çok güçlü bir dil. Bu gücü veren harika modüller var bunlardan biri de **Requests modülü**.

Bu modül ile web üzerindeki isteklerinizi yönetebilirsiniz. Mesela bu modül ile API entpointlerine PUT, DELETE, POST gibi istekler atabilirsiniz.



Requests modülü ile ilgili youtube kanalında bir video hazırladım ve buraya ekledim. İlk önce yazıları okuyup sonra videoyu izlemenizde fayda var.



Kurulum

Ben pip3 ile kurmuştum, pip3 python3 için paket yöneticisi.

```
sudo apt-get install python3-pip
```

komutu ile pip3 kullanmaya başlayabiliriz. Eğer pip3 yüklüyse buna gerek yok, şimdi requests modülünü kuralım.

```
pip3 install requests
```

Artık modülü kurduk, projemiz içerisine

```
import requests
```

diyerek aktaralım. Şimdi de bu güzel modülün özelliklerine geçelim.

Not: Bütün örneklerde <http://httpbin.org/> adresini kullanacağım, bu adres HTTP isteği yapıp cevap alabileceğiniz bir servis.

HTTP İstekleri

```
>>> r = requests.get('http://httpbin.org/get')
>>> r = requests.post('http://httpbin.org/post')
>>> r = requests.put('http://httpbin.org/put')
>>> r = requests.delete('http://httpbin.org/delete')
```

Bu şekilde istekler gönderebiliyoruz. Elbette ki istek göndermek ile olmuyor, parametre gönderme ihtiyacı var. Şimdi istek atarken nasıl parametre gönderebileceğimize bakalım.

Parametre Göndermek

Parametre göndermek için params sözlüğünü kullanıyoruz. Hemen bir örnek verelim.

```
>>> r = requests.get('http://httpbin.org/get', params={"kategori": "elektronik", "marka": "samsung"})
>>> r.url
'http://httpbin.org/get?kategori=elektronik&marka=samsung'
```

params sözlüğü içinde parametre geçtim, **url** methodu ile de istek yapılan url adresini görebilirsiniz. Dikkat ettiyseniz verdiğimiz parametreler bağlantı adresinin sonuna eklendi.

Bazen istek attığınız sayfa başka bir sayfaya yönlendiriyor olabilir, bu yönlendirmeyi takip edebilir veya etmeyebiliriz, bunun için de `allow_redirects=True` veya `allow_redirects=False` kullanıyoruz.

```
>>> r = requests.get("http://httpbin.org/redirect/1", allow_redirects=False)
>>> r.status_code
302
>>> r = requests.get("http://httpbin.org/redirect/1", allow_redirects=True)
>>> r.status_code
200
```

Dikkat ederseniz `allow_redirects=False` yapıldığında yönlendirmeyi kapattı ve o sayfanın `status_code` değeri 302 döndü. True yapınca da yönlendirmeye izin verildi ve `status_code` 200 döndü. HTTP status

kodlarını incelemek isterseniz [buraya](#) tıklayabilirsiniz.

Bu örnek GET isteği iindi, imdi bir de POST iin sanki bir HTML formu doldurmu ve o bilgileri post etmiiz gibi davranalım.

```
>>> r = requests.post("http://httpbin.org/post", data={"username": "sinan", "password": "asd123"})
>>> r.status_code
200
>>> r.json()
{'files': {}, 'args': {}, 'url': 'http://httpbin.org/post', 'json': None, 'data': ''}
>>> r.json()["form"]
{'username': 'sinan', 'password': 'asd123'}
```

Dikkat ettiyseniz data parametresi ile username ve password bilgilerini formdata olarak gnderdim. Ardından **status.code** methodu ile 200 yani baarılı olduėunu kontrol ettim. Ardından **json** methodu ile dnen deėeri ekrana bastırdım. Kullandıėımız httpbin.org servisi, post ettiėimiz parametreleri bize geri dndryor. Bu nedenle json iinden sadece form alanını ekerek gnderdiėim parametreleri grebildim.

İstek atarken timeout deėeri de belirleyebilirsiniz, saniye olarak bir deėer atayabilirsiniz.

```
>>> r = requests.get("http://httpbin.org/get", timeout=1)
```

Belirtilen timeout iinde cevap alamaz ise hata verecektir.

Burada bir parantez aalım, REST API ile alııyorsak, farklı endpointlere json post etmemiz gerekecek. Bu noktada da hemen json modl bize yardım ediyor. Basit bir rnek yapalım ve bu konuyu bitirelim.

```
>>> import json
>>> import requests
>>> endpoint = "http://httpbin.org/post"
>>> myData = {
...     "id": 1,
...     "name": "Leanne Graham",
...     "username": "Bret",
...     "email": "Sincere@april.biz",
...     "address": {
...         "street": "Kulas Light",
...         "suite": "Apt. 556",
...         "city": "Gwenborough",
...         "zipcode": "92998-3874",
...         "geo": {
...             "lat": "-37.3159",
...             "lng": "81.1496"
...         }
...     },
...     "phone": "1-770-736-8031 x56442",
...     "website": "hildegard.org",
...     "company": {
...         "name": "Romaguera-Crona",
...         "catchPhrase": "Multi-layered client-server neural-net",
...         "bs": "harness real-time e-markets"
...     }
... }
>>> r = requests.post(endpoint, data=json.dumps(myData))
>>> r.status_code
200
```

zel Header Kullanımı

İstek atarken, headers parametresi ile szlk formatında istediėiniz bilgileri girebilirsiniz.

```
>>> r = requests.post("http://httpbin.org/post", headers={"User-Agent": "Sinan-Chrome"})
>>> r.status_code
200
```

Bu rnek ierisinde headers parametresi ile zel bir user-agent deėeri gndermi oldum. **status.code** methodu ile de isteėe dnen durum kodunu kontrol ettim. Benim rneėimde 200 gelmi, yani baarılı.

İstek attıktan sonra kullanılan methodlar

Tm rneklerde bir r deėikenine aktarmıtık attıėımız tm istekleri, imdi bu r deėikeni iinde yani requests modl iindeki istek attıktan sonra kullanabileceėimiz methodları inceleyelim.

text

Sitenin HTML ieriėini dndrr.

```
>>> r = requests.get("http://httpbin.org/")
>>> r.text
```

headers

Header bilgilerini gsterir.

```
>>> r = requests.get("http://httpbin.org/")
>>> r.headers
{'Content-Length': '13011', 'Content-Type': 'text/html; charset=utf-8', 'Server': 'm...}
```

Dilerseniz istediėiniz header bilgilerinizi siz get isteėi ile birlikte belirtebilirsiniz.

```
r = requests.get("http://httpbin.org/", headers={'user-agent': 'sinanerdinc'})
```

url

Hangi url adresine istek gönderdiğini döner. İstek yaparken parametre olarak bazı değerler gönderebilirsiniz.

```
>>> r = requests.get("http://httpbin.org/", params={"ad":"sinan", "soyad":"erdinc"})
>>> r.url
'http://httpbin.org/?soyad=erdinc&ad=sinan'
```

Url adresinin sonunda verdiğim parametrelerin nasıl iliştilirildiğine dikkat edin.

status_code

İsteğin HTTP durum kodunu döndürür.

```
>>> r = requests.get("http://httpbin.org/")
>>> r.status_code
200
```

history

Bir istek attınız ve status_code değeri 200 geldi. Ancak belki 2 kere 301 sonra 1 kere 302 yönlendirme ile en son bir sayfaya geldi ve 200 döndü. Buna bakabilmek için bu method kullanılıyor.

```
>>> r = requests.get("http://httpbin.org/redirect/4", allow_redirects=True)
>>> r.history
[<Response [302]>, <Response [302]>, <Response [302]>, <Response [302]>]
```

http://httpbin.org/redirect/4 adresi 4 kere 302 yönlendirmesi yapıyor. İşte buraya istek atınca history methodu ile de kontrol edince geçmişteki HTTP durum kodlarını görebiliyoruz.

encoding

Sayfanın encode değerini döner.

```
>>> r = requests.get("http://github.com", timeout=1)
>>> r.encoding
'utf-8'
```

request

Yaptığınız isteğin ne olduğunu döner, GET, POST v.s.

```
>>> r.request
<PreparedRequest [GET]>
>>> r.request.method
'GET'
```

elapsed

Geçen zamanı döner.

```
>>> r.elapsed.total_seconds()
0.566292
```



← ÖNCEKİ YAZI

SONRAKİ YAZI →

sinanerdinc.com Yorum likesi

Yorum yazmak, bu konu hakkında özgürce fikirlerinizi paylaşmanıza olanak tanır.



0 Yorum sinanerdinc.com Disqus Gizlilik İlkesi 1 Oturum Açın

Öner Tweet Gönder Paylaş En İyiye Göre Sırala



Tartışma başlat...

ŞUNUNLA OTURUM AÇIN

YA DA BİR DISQUS HESABI AÇIN



Ad

İlk yorum yapan siz olun.

Abone Olun Sitenize Disqus ekleyin Verilerim Satılmısn

DISQUS



Sinan Erdinc • 2020 • sinanerdinc.com

sitemap