

```
In [2]: import requests
#yüklenmezse aşağıdakini yaz
#pip install requests
```

parametre istemeyen

```
In [13]: bolgeler_URL = "https://data.police.uk/api/forces" # 200 verecek çünkü ulaşılabilir bi site
requests.get(bolgeler_URL)
```

Out [13]: <Response [200]>

```
In [9]: bolgeler_URL = "https://data.police.uk/api/force" #linkte forces > force yaparsak sayfaya ulaşamayacak ve 404 verecek
requests.get(bolgeler_URL)
```

Out [9]: <Response [404]>

```
In [14]: response = requests.get(bolgeler_URL)
response.status_code
```

Out [14]: 200

```
In [15]: response.url
```

Out [15]: 'https://data.police.uk/api/forces'

```
In [16]: response.text
```

```
Out [16]: [{"id":"avon-and-somerset","name":"Avon and Somerset Constabulary"}, {"id":"bedfordshire","name":"Bedfordshire Police"}, {"id":"cambridgeshire","name":"Cambridgeshire Constabulary"}, {"id":"cheshire","name":"Cheshire Constabulary"}, {"id":"city-of-london","name":"City of London Police"}, {"id":"cleveland","name":"Cleveland Police"}, {"id":"cumbria","name":"Cumbria Constabulary"}, {"id":"derbyshire","name":"Derbyshire Constabulary"}, {"id":"devon-and-cornwall","name":"Devon & Cornwall Police"}, {"id":"dorset","name":"Dorset Police"}, {"id":"durham","name":"Durham Constabulary"}, {"id":"dyfed-powys","name":"Dyfed-Powys Police"}, {"id":"essex","name":"Essex Police"}, {"id":"gloucestershire","name":"Gloucestershire Constabulary"}, {"id":"greater-manchester","name":"Greater Manchester Police"}, {"id":"gwent","name":"Gwent Police"}, {"id":"hampshire","name":"Hampshire Constabulary"}, {"id":"hertfordshire","name":"Hertfordshire Constabulary"}, {"id":"humberside","name":"Humberside Police"}, {"id":"kent","name":"Kent Police"}, {"id":"lancashire","name":"Lancashire Constabulary"}, {"id":"leicestershire","name":"Leicestershire Police"}, {"id":"lincolnshire","name":"Lincolnshire Police"}, {"id":"merseyside","name":"Merseyside Police"}, {"id":"metropolitan","name":"Metropolitan Police Service"}, {"id":"norfolk","name":"Norfolk Constabulary"}, {"id":"north-wales","name":"North Wales Police"}, {"id":"north-yorkshire","name":"North Yorkshire Police"}, {"id":"northamptonshire","name":"Northamptonshire Police"}, {"id":"northumbria","name":"Northumbria Police"}, {"id":"nottinghamshire","name":"Nottinghamshire Police"}, {"id":"northern-ireland","name":"Police Service of Northern Ireland"}, {"id":"south-wales","name":"South Wales Police"}, {"id":"south-yorkshire","name":"South Yorkshire Police"}, {"id":"staffordshire","name":"Staffordshire Police"}, {"id":"suffolk","name":"Suffolk Constabulary"}, {"id":"surrey","name":"Surrey Police"}, {"id":"sussex","name":"Sussex Police"}, {"id":"thames-valley","name":"Thames Valley Police"}, {"id":"warwickshire","name":"Warwickshire Police"}, {"id":"west-mercia","name":"West Mercia Police"}, {"id":"west-midlands","name":"West Midlands Police"}, {"id":"west-yorkshire","name":"West Yorkshire Police"}, {"id":"wiltshire","name":"Wiltshire Police"}]
```

```
In [18]: response.json() # response.text in okunaklı hali, dictionary verir
```

```
Out [18]: [{'id': 'avon-and-somerset', 'name': 'Avon and Somerset Constabulary'}, {'id': 'bedfordshire', 'name': 'Bedfordshire Police'}, {'id': 'cambridgeshire', 'name': 'Cambridgeshire Constabulary'}, {'id': 'cheshire', 'name': 'Cheshire Constabulary'}, {'id': 'city-of-london', 'name': 'City of London Police'}, {'id': 'cleveland', 'name': 'Cleveland Police'}, {'id': 'cumbria', 'name': 'Cumbria Constabulary'}, {'id': 'derbyshire', 'name': 'Derbyshire Constabulary'}, {'id': 'devon-and-cornwall', 'name': 'Devon & Cornwall Police'}, {'id': 'dorset', 'name': 'Dorset Police'}, {'id': 'durham', 'name': 'Durham Constabulary'}, {'id': 'dyfed-powys', 'name': 'Dyfed-Powys Police'}, {'id': 'essex', 'name': 'Essex Police'}, {'id': 'gloucestershire', 'name': 'Gloucestershire Constabulary'}, {'id': 'greater-manchester', 'name': 'Greater Manchester Police'}, {'id': 'gwent', 'name': 'Gwent Police'}, {'id': 'hampshire', 'name': 'Hampshire Constabulary'}, {'id': 'hertfordshire', 'name': 'Hertfordshire Constabulary'}, {'id': 'humberside', 'name': 'Humberside Police'}, {'id': 'kent', 'name': 'Kent Police'}, {'id': 'lancashire', 'name': 'Lancashire Constabulary'}, {'id': 'leicestershire', 'name': 'Leicestershire Police'}, {'id': 'lincolnshire', 'name': 'Lincolnshire Police'}, {'id': 'merseyside', 'name': 'Merseyside Police'}, {'id': 'metropolitan', 'name': 'Metropolitan Police Service'}, {'id': 'norfolk', 'name': 'Norfolk Constabulary'}, {'id': 'north-wales', 'name': 'North Wales Police'}, {'id': 'north-yorkshire', 'name': 'North Yorkshire Police'}, {'id': 'northamptonshire', 'name': 'Northamptonshire Police'}, {'id': 'northumbria', 'name': 'Northumbria Police'}, {'id': 'nottinghamshire', 'name': 'Nottinghamshire Police'}, {'id': 'northern-ireland', 'name': 'Police Service of Northern Ireland'}, {'id': 'south-wales', 'name': 'South Wales Police'}, {'id': 'south-yorkshire', 'name': 'South Yorkshire Police'}, {'id': 'staffordshire', 'name': 'Staffordshire Police'}, {'id': 'suffolk', 'name': 'Suffolk Constabulary'}, {'id': 'surrey', 'name': 'Surrey Police'}, {'id': 'sussex', 'name': 'Sussex Police'}, {'id': 'thames-valley', 'name': 'Thames Valley Police'}, {'id': 'warwickshire', 'name': 'Warwickshire Police'}, {'id': 'west-mercia', 'name': 'West Mercia Police'}, {'id': 'west-midlands', 'name': 'West Midlands Police'}, {'id': 'west-yorkshire', 'name': 'West Yorkshire Police'}, {'id': 'wiltshire', 'name': 'Wiltshire Police'}]
```

parametre isteyen

```
In [36]: bolgeler_URL = "https://data.police.uk/docs/method/crimes-no-location/"
requests.get(bolgeler_URL)
```

Out [36]: <Response [200]>

```
In [37]: sucKategorileri_URL = "https://data.police.uk/api/crime-categories?date=2011-08" # ?xxxx= şeklinde olan url ler parametre a
payload = {"date": "2024-02"} # burada dictionary tipinde giriş yapmamız gerekiyor
response = requests.get(sucKategorileri_URL, params=payload)
```

[illegible]

```
response1.json
```

```
In [87]: response1.json() #vermiyor çünkü data html türünde json değil sanırım güncellemişler
```

```
In [77]: # AUTHOR: YAVUZ SELIM BASTEMUR
# ASSISTED BY JR. DEV. SAMET ENDS ORSDEMIR
# TOTAL TIME SPENT: 17 minutes (Including as coffee break :D)
import requests
from collections import Counter

class SucRaporu():
    def __init__(self, bolge, tarih, suc_tipi="all-crime"):
        self.bolge = bolge
        self.tarih = tarih
        self.suc_tipi = suc_tipi
        self.suclar = self.suclari_bul()

    def url_hazirla(self):
        urlPart1="https://data.police.uk/api/crimes-no-location?category="
        urlPart2="&force="
        urlPart3="&date="
        return "".join([
            urlPart1,
            str(self.suc_tipi),
            urlPart2,
            str(self.bolge),
            urlPart3,
            str(self.tarih)])

    def suclari_bul(self):
        #suc_URL = "https://data.police.uk/docs/method/crimes-no-location"
        enes_url = "https://data.police.uk/api/crimes-no-location?category=all-crime&force=norfolk&date=2023-11" # Buradaki
        # Aşağıdaki payload doğru bir şekilde doldurulmamış. "params=" kullanılmış mantıken olması lazım ama anasayfanın
        # 0 yüzden url yi elle yazdim anadim ki sorun çözüldü. Ama bu programlama mantigana aykiriidir. Bunun çözülmesi gere
        # Çözüm yöntemi olarak string manipulasyonu yapılabilir. Örneğin: new url - sue_url + ? + category payload("categ
```

```

        payload = {
            "category": self.suc_tipi,
            "force": self.bolge,
            "date": self.tarih}

        #response = requests.get (suc_URL, params= payload)
        response = requests.get(enes_url)
        if response.status_code==200:
            return response.json()
        else:
            return None

    def suclari_raporla(self):
        suclar_listesi=[]

        if self.suclar is not None:
            for suc in self.suclar:
                suclar_listesi.append(suc["category"])

            return Counter(suclar_listesi)

```

```
In [78]: sr=SucRaporu("norfolk","2020-11")
```

```
In [79]: print(sr.suclari_raporla())
```

```
Counter({'violent-crime': 36, 'other-theft': 14, 'public-order': 13, 'shoplifting': 13, 'drugs': 12, 'criminal-damage-arson': 9, 'burglary': 4, 'vehicle-crime': 3,
```

```
In [80]: #hatta daha iyisini yaptık
```

```
In [81]: import requests
from collections import Counter

class SucRaporu():
    def __init__(self, suc_tipi, bolge, tarih):
        self.suc_tipi = suc_tipi
        self.bolge = bolge
        self.tarih = tarih

    def url_hazirla(self):
        urlPart1 = "https://data.police.uk/api/crimes-no-location?category="
        urlPart2 = "&force="
        urlPart3 = "&date="
        enes_url2 = urlPart1 + self.suc_tipi + urlPart2 + self.bolge + urlPart3 + self.tarih
        return enes_url2

    def suclari_bul(self):
        enes_url2 = self.url_hazirla()
        response = requests.get(enes_url2)
        print(f"Request URL: {enes_url2}")
        print(f"Response Status Code: {response.status_code}")

        if response.status_code == 200:
            return response.json()
        else:
            print(f"Error: {response.text}")
            return None

    def suclari_raporla(self):
        suclar_listesi = []
        self.suclar = self.suclari_bul()
        if self.suclar is not None and isinstance(self.suclar, list):
            for suc in self.suclar:
                suclar_listesi.append(suc["category"])
            rapor = Counter(suclar_listesi)

            if rapor:
                sorted_rapor = sorted(rapor.items())
                rapor_text = "\n".join([f"{suç}: {sayı}" for suç, sayı in sorted_rapor])
                return rapor_text
            else:
                return "Herhangi bir suç bulunamadı."
        else:
            return "Veri alınamadı."

```

```
In [82]: sr = SucRaporu("all-crime", "norfolk", "2023-11")
```

```
In [84]: rapor = sr.suclari_raporla()
```

```
Request URL: https://data.police.uk/api/crimes-no-location?category=all-crime&force=norfolk&date=2023-11
Response Status Code: 200
```

```
In [85]: print(rapor)
```

```
bicycle-theft: 2
burglary: 4
criminal-damage-arson: 9
drugs: 12
other-theft: 14
possession-of-weapons: 2
public-order: 13
robbery: 2
shoplifting: 13
vehicle-crime: 3
violent-crime: 36
```

```
In [86]: sr2 = SucRaporu("all-crime", "norfolk", "2020-11")
```

```
In [87]: rapor = sr2.suclari_raporla()
```

```
Request URL: https://data.police.uk/api/crimes-no-location?category=all-crime&force=norfolk&date=2020-11
Response Status Code: 404
Error:
```

```
In [88]: print(rapor)
```

Veri alınamadı.

```
In [ ]:
```