

Uygulama Programı Arayüzü

Bu Haftanın Ders Kazanımları

1

API Get

2

API Post

3

API Put - Patch

4

API Delete

API Get



Response – Json Dönüşüm

```
import json
import requests
response = requests.get("https://furkandurmus.com/tools/api/users")
users = json.loads(response.text)
```



```
import requests
response = requests.get("https://furkandurmus.com/tools/api/users")
users = response.json()
```

json modülünü import etmeye gerek kalmadığı ve yanıtı doğrudan Python veri tipine çevirdiği için response.json() kullanmak çok daha pratik ve temizdir.



Örnek 1: Api Döngü Kullanımı

```
import requests  
  
response = requests.get("https://furkandurmus.com/tools/api/users")
```

Üstteki api adresine istek atarak, alttaki çıktıyı elde edecek kodu yazınız.

```
Öğrenci Sayısı: 5  
Şehirler:  
1: Samsun  
2: Ankara  
3: İstanbul  
4: İzmir  
5: Konya
```



Örnek 1: Api Döngü Kullanımı

```
import requests

response = requests.get("https://furkandurmus.com/tools/api/users")

users = response.json()

print("Öğrenci Sayısı:", len(users))
print("Şehirler:")
i = 1
for user in users:
    print(f"{i}: {user['address']['city']}")
    i += 1
```

Üstteki api adresine istek atarak, alttaki çıktıyı elde edecek kodu yazınız.

```
Öğrenci Sayısı: 5
Şehirler:
1: Samsun
2: Ankara
3: İstanbul
4: İzmir
5: Konya
```



Örnek 2 : Api Filtreleme

Verilen API adresinden kullanıcı verilerini çekiniz. Kullanıcıdan alınan ID bilgisine göre filtreleme yaparak eşleşen kaydı ekrana yazdırınız; eşleşen kayıt bulunamazsa ekrana 'not found' yazdırınız.

```
import requests
```

```
response = requests.get("https://furkandurmus.com/tools/api/users")
```

```
Enter user id: 3
```

```
{'id': 3, 'name': 'Zeynep Kaya', 'username': 'zeynepk', 'email': 'zeynep@example.com', 'phone': '0555 333 3 33', 'website': 'zeynepkaya.test', 'address': {'street': 'İnönü Caddesi', 'suite': 'No: 8', 'city': 'İstanbul', 'zipcode': '34000'}, 'company': {'name': 'Marmara Dijital', 'catchPhrase': 'Web için yaratıcı fikirler'}}
```

```
Enter user id: 11
User not found
```

Örnek 2 : Api Filtreleme

Verilen API adresinden kullanıcı verilerini çekiniz. Kullanıcıdan alınan ID bilgisine göre filtreleme yaparak eşleşen kaydı ekrana yazdırınız; eşleşen kayıt bulunamazsa ekrana 'not found' yazdırınız.

```
import requests

response = requests.get("https://furkandurmus.com/tools/api/users")

users = response.json()
id = int(input("Enter user id: "))
found = False

for user in users:
    if user["id"] == id:
        found = True
        print(user)
        break

if found == False:
    print("User not found")
```

```
Enter user id: 3
{'id': 3, 'name': 'Zeynep Kaya', 'username': 'zeynepk', 'email': 'zeynep@example.com', 'phone': '0555 333 3
3 33', 'website': 'zeynepkaya.test', 'address': {'street': 'İnönü Caddesi', 'suite': 'No: 8', 'city': 'İsta
nbul', 'zipcode': '34000'}, 'company': {'name': 'Marmara Dijital', 'catchPhrase': 'Web için yaratıcı fikirl
er'}}
```

```
Enter user id: 11
User not found
```

Örnek 3 : Api Veri Bölme

Yandaki posts API adresindeki veriler içerisinde 30 like üzeri verileri localde bir listeye kayıt ediniz.

```
import requests

response = requests.get("https://furkandurmus.com/tools/api/posts")
```

```
[{'userId': 2, 'id': 6, 'like': 33, 'title': 'DELETE isteği', 'body': 'DELETE metodu belirli bir kaynağın silinmesini istemek için kullanılır.'}, {'userId': 3, 'id': 7, 'like': 46, 'title': 'HTTP durum kodları', 'body': '200 başarılı istekleri, 201 yeni kaynak oluşturmayı, 404 ise kaynağın bulunamadığını ifade eder.'}, {'userId': 3, 'id': 8, 'like': 67, 'title': 'JSON veri biçimi', 'body': 'JSON, anahtar ve değer çiftleriyle veri taşımak için sık kullanılan metin tabanlı bir formattır.'}, {'userId': 3, 'id': 9, 'like': 42, 'title': 'REST kaynakları', 'body': 'REST yaklaşımında URL adresleri çoğunlukla kullanıcılar, yazılar veya görevler gibi kaynakları temsil eder.'}]
```

Örnek 3 : Api Veri Bölme

Yandaki posts API adresindeki veriler içerisinde 30 like üzeri verileri localde bir listeye kayıt ediniz.

```
import requests

response = requests.get("https://furkandurmus.com/tools/api/posts")
posts = response.json()

best_posts = []

for post in posts:
    if post["like"] > 30:
        best_posts.append(post)

print(best_posts)
```

Örnek 4 : Api Filtre + Veri Bölme

Yandaki posts API adresindeki veriler içerisinde 30 like üzeri verilerin isimlerini localde bir listeye kayıt ediniz.

```
import requests

response = requests.get("https://furkandurmus.com/tools/api/posts")
```

```
['DELETE isteği', 'HTTP durum kodları', 'JSON veri biçimi', 'REST kaynakları']
```

Örnek 4 : Api Filtre + Veri Bölme

Yandaki posts API adresindeki veriler içerisinde 30 like üzeri verilerin isimlerini localde bir listeye kayıt ediniz.

Bu derste yaptığımız filtering, API'nin kendi filtering özelliği değildir. API'den gelen verinin tamamını aldıktan sonra Python tarafında yaptığımız filtrelemedir.

```
import requests

response = requests.get("https://furkandurmus.com/tools/api/posts")
posts = response.json()

best_posts = []

for post in posts:
    if post["like"] > 30:
        best_posts.append(post["title"])

print(best_posts)
```

```
['DELETE isteği', 'HTTP durum kodları', 'JSON veri biçimi', 'REST kaynakları']
```

Örnek 5 : Yeni Sözlük

Yandaki posts API adresindeki veriler içerisinde 30 like üzeri verilerin isimlerini ve like sayılarını localde bir listeye kayıt ediniz.

```
import requests

response = requests.get("https://furkandurmus.com/tools/api/posts")
```

```
[{'baslik': 'DELETE isteği', 'begeni': 33}, {'baslik': 'HTTP durum kodları', 'begeni': 46}, {'baslik': 'JSON veri biçimi', 'begeni': 67}, {'baslik': 'REST kaynakları', 'begeni': 42}]
```

Örnek 5 : Yeni Sözlük

Yandaki posts API adresindeki veriler içerisinde 30 like üzeri verilerin isimlerini ve like sayılarını localde bir listeye kayıt ediniz.

```
import requests

response = requests.get("https://furkandurmus.com/tools/api/posts")
posts = response.json()

best_posts = []

for post in posts:
    if post["like"] > 30:
        veri = {
            "baslik": post["title"],
            "begeni": post["like"]
        }
        best_posts.append(veri)

print(best_posts)
```

```
[{'baslik': 'DELETE isteği', 'begeni': 33}, {'baslik': 'HTTP durum kodları', 'begeni': 46}, {'baslik': 'JSON veri biçimi', 'begeni': 67}, {'baslik': 'REST kaynakları', 'begeni': 42}]
```

API Post



Giriş

API POST, bir istemcinin (tarayıcı, mobil uygulama veya Python script'i) sunucuya yeni bir veri göndermek veya yeni bir kaynak oluşturmak için kullandığı temel HTTP metodudur.

Kısaca: GET sunucudan veri çekmek için kullanılırken, POST sunucuya veri yüklemek/kaydetmek için kullanılır.



Giriş

Ne Zaman Kullanılır?

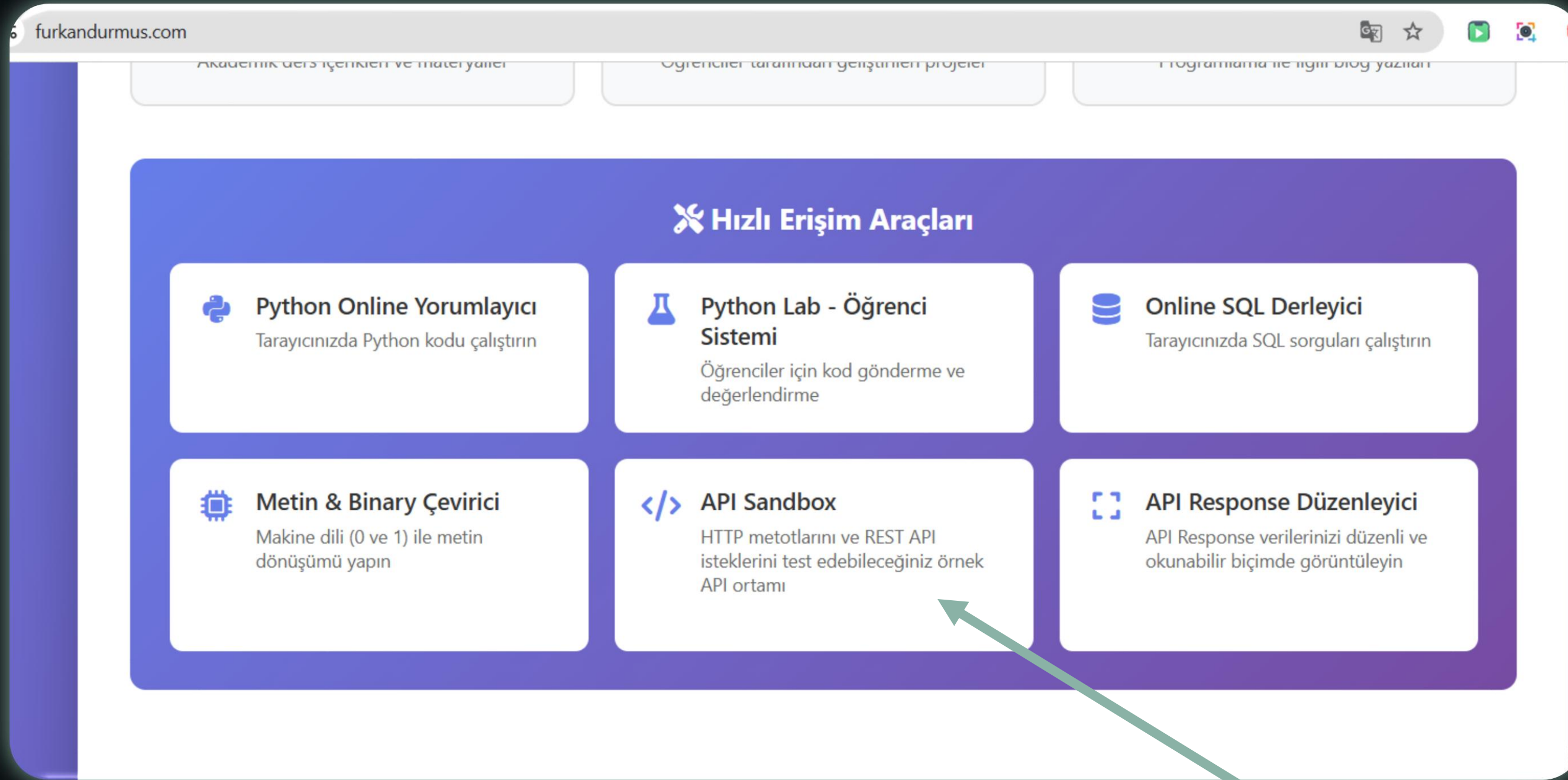
- Bir web sitesine yeni üye kaydı (register) yaparken,
- Bir forma girilen bilgileri (iletişim formu, anket vb.) kaydederken,
- Bir sosyal medya platformuna yeni bir gönderi (post), yorum veya fotoğraf eklerken,
- Sisteme kullanıcı adı ve şifre ile giriş (login) yaparken (verilerin URL'de görünmemesi için).

Genel olarak bir api'ye yeni veri göndermek istediğimiz her durumda api post kullanılır.



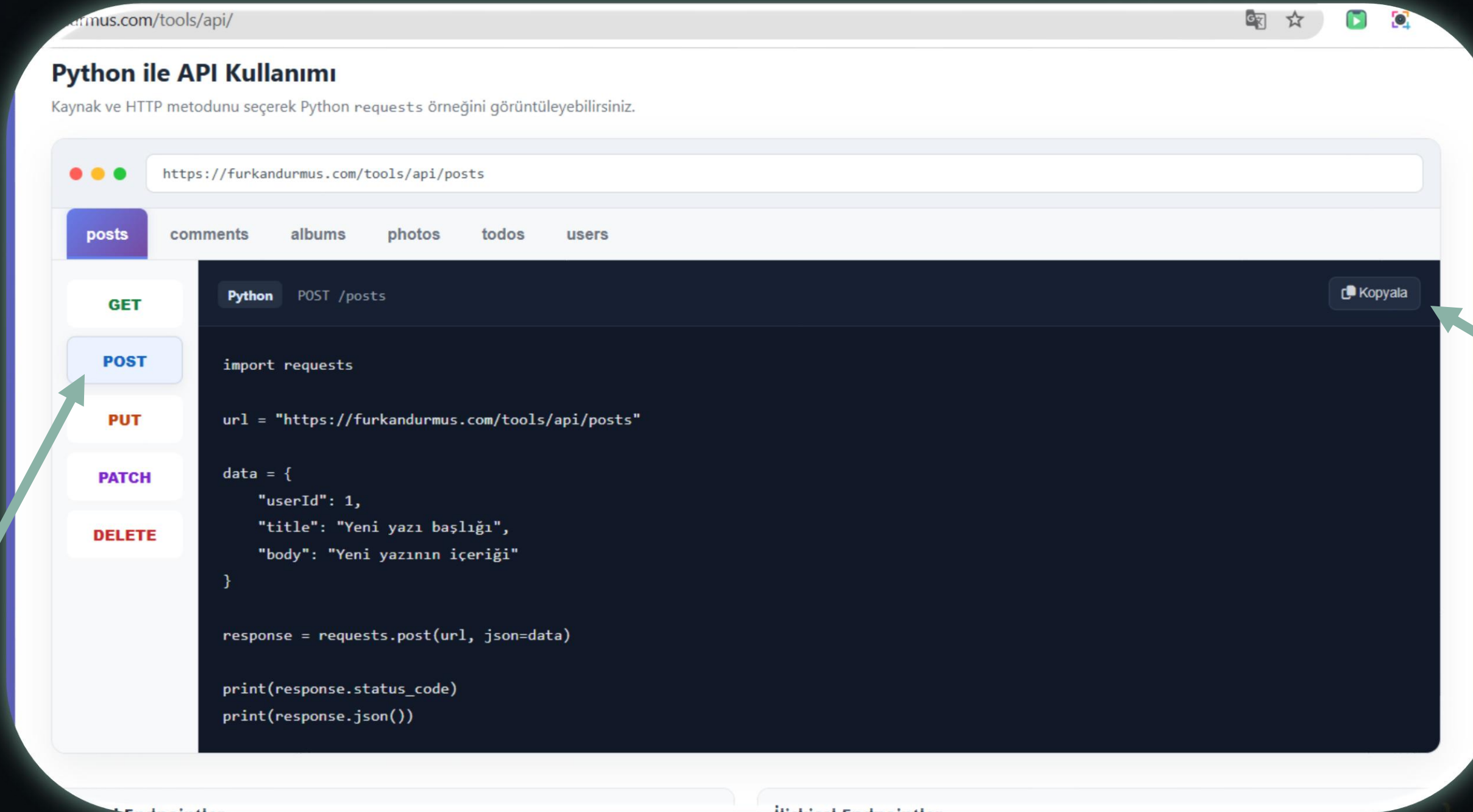
Giriş

Furkandurmus.com -> API Sandbox sayfasını açınız.



Giriş

Post sekmesine tıklayıp ilgili python kodunu bilgisayarınızda çalıştırınız.



Giriş

Sayfadaki genel datanın bozulmaması için server tarafında yapılan güncellemeler anlıktır. Size başarılı olmuş gibi response verir fakat ana data değişmez. Gerçek API'lerde post istekleri gerçekten ana dataya veri ekler.

```
201
```

```
{'userId': 1, 'title': 'Yeni yazı başlığı', 'body': 'Yeni yazının içeriği', 'id': 16}
```

Sandbox davranışı

POST, PUT, PATCH ve DELETE istekleri başarılı olmuş gibi yanıt üretir; ancak ana veri kümesi değiştirilmez. Aynı kaynağa daha sonra GET isteği gönderildiğinde başlangıçtaki veri tekrar döner.

Giriş

Post işleminde data 'json' anahtar isimlendirmesiyle gönderilir. Buradaki json ismi sabit data ise bizim datamızı oluşturduğumuz değişken ismidir.

```
data = {  
    "userId": 1,  
    "title": "Yeni yazı başlığı",  
    "body": "Yeni yazının içeriği"  
}
```

```
response = requests.post(url, json=data)
```

Örnek

<https://furkandurmus.com/tools/api/todos> adresinden önce bir get isteği ile api yapısını inceleyiniz. Ardından aynı adrese bir post verisi gönderen kodu yazınız.

Örnek

<https://furkandurmus.com/tools/api/todos> adresinden önce bir get isteği ile api yapısını inceleyiniz. Ardından aynı adrese bir post verisi gönderen kodu yazınız.

```
import requests

url = "https://furkandurmus.com/tools/api/todos"

data = {
    "userId": 1,
    "title": "Yeni görev",
    "completed": False,
}

response = requests.post(url, json=data)

print(response.status_code)
print(response.json())
```

Örnek

Furkandurmus.com/tools/api baseurl'inde bulunan endpointler nesne tipinde veri beklemektedirler. Localde bir nesne listesini oluşturunuz ve uygun formatta bunu users endpoint'ine iletiniz.

```
users = [  
  { ..... },  
  { ..... },  
  { ..... }  
]
```

Örnek

Furkandurmus.com/tools/api baseurl'inde bulunan endpointler nesne tipinde veri beklemektedirler. Localde bir nesne listesin oluşturunuz ve uygun formatta bunu users endpoint'ine iletiniz.

```
import requests

url = "https://furkandurmus.com/tools/api/users"

users = [{
    "name": "Yeni Kullanıcı",
    "username": "yenikullanici",
    "email": "kullanici@example.com"
},
{
    "name": "Yeni Kullanıcı 2",
    "username": "yenikullanici2",
    "email": "kullanici2@example.com"
},
{
    "name": "Yeni Kullanıcı 3",
    "username": "yenikullanici3",
    "email": "kullanici3@example.com"
}]

for user in users:
    response = requests.post(url, json=user)
    print(response.status_code)
    print(response.json())
```

API PUT - PATCH ➔



Örnek

<https://furkandurmus.com/tools/api/users/1> endpointindeki veriyi get inceleyiniz ve bu adresteki name ve surname alanını put ile güncelleyiniz.

```
response = requests.put(url, json=data)
```



Örnek

<https://furkandurmus.com/tools/api/users/1> endpointindeki veriyi get inceleyiniz ve bu adresteki name ve surname alanını put ile güncelleyiniz.

```
import requests

2
3 url = "https://furkandurmus.com/tools/api/users/1"
4
5 data = {
6     "name": "Yeni Kullanıcı",
7     "username": "yenikullanici",
8 }
9
10
11 response = requests.put(url, json=data)
12
13 print(response.status_code)
14 print(response.json())
```

200

```
{'name': 'Yeni Kullanıcı', 'username': 'yenikullanici', 'id': 1}
```

Örnek

<https://furkandurmus.com/tools/api/users/1> endpointindeki veriyi get inceleyiniz ve bu adresteki name ve surname alanını patch ile güncelleyiniz ve çıktığı yorumlayınız.

```
response = requests.patch(url, json=data)
```



Örnek

<https://furkandurmus.com/tools/api/users/1> endpointindeki veriyi get inceleyiniz ve bu adresteki name ve surname alanını patch ile güncelleyiniz ve çıktığı yorumlayınız.

```
import requests

url = "https://furkandurmus.com/tools/api/users/1"

data = {
    "name": "Yeni Kullanıcı",
    "username": "yenikullanici",
}

response = requests.patch(url, json=data)

print(response.status_code)
print(response.json())
```

```
200
{'id': 1, 'name': 'Yeni Kullanıcı', 'username': 'yenikullanici', 'email': 'ayse@example.com', 'phone': '0555 111 11 11', 'website': 'ayseyilmaz.test', 'address': {'street': 'Atatürk Bulvarı', 'suite': 'No: 12', 'city': 'Samsun', 'zipcode': '55000'}, 'company': {'name': 'Kuzey Yazılım', 'catchPhrase': 'Basit, hızlı ve güvenilir çözümler'}}
```

Örnek

<https://furkandurmus.com/tools/api/comments/1> endpointindeki veriyi get inceleyiniz ve bu adresteki istediğiniz alanları put ile güncelleyiniz.

```
response = requests.put(url, json=data)
```



Örnek

<https://furkandurmus.com/tools/api/comments/1>
endpointindeki veriyi get
inceleyiniz ve bu adresteki
istediğiniz alanları put ile
güncelleyiniz.

```
1 import requests
2
3 url = "https://furkandurmus.com/tools/api/comments/1"
4
5 data = {
6     "postId": 1,
7     "name": "Yeni yorum",
8     "email": "ogrenci@example.com",
9     "body": "Yorum içeriği"
10 }
11
12 response = requests.put(url, json=data)
13
14 print(response.status_code)
15 print(response.json())
```

200

```
{'postId': 1, 'name': 'Yeni yorum', 'email': 'ogrenci@example.com', 'body': 'Yorum içeriği', 'id': 1}
```

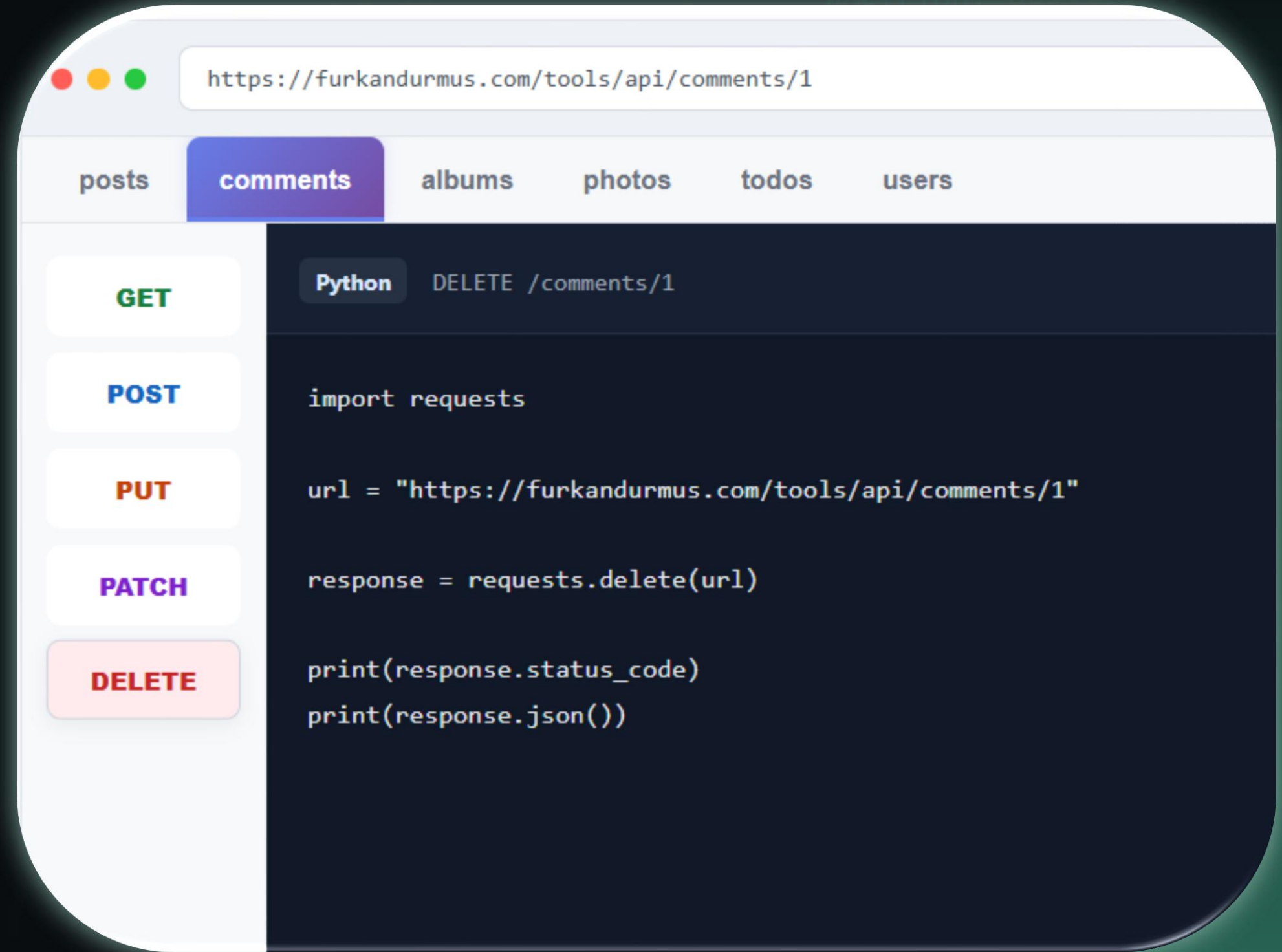
API Delete



İnceleme

Web sitesinden Örnek delete komutlarını bilgisayarınızda çalıştırınız.

```
response = requests.delete(url)
```



Örnek

<https://furkandurmus.com/tools/api/posts>

Ve

<https://furkandurmus.com/tools/api/comments> adreslerine istek atınız ve yorumlayınız.

Örnek

<https://furkandurmus.com/tools/api/posts>

<https://furkandurmus.com/tools/api/comments>

```
{
  "userId": 1,
  "id": 1,
  "like": 12,
  "title": "API nedir?",
  "body": "API, farklı yazılım bileşenlerinin belirli kurallar üzerinden haberleşmesini sağlayan bir arayüzdür."
},
{
  "userId": 1,
  "id": 2,
  "like": 24,
  "title": "HTTP ile ilk adım",
  "body": "İstemci sunucuya bir istek gönderir, sunucu da isteği işleyerek uygun bir yanıt döndürür."
},
{
  "userId": 1,
  "id": 3,
  "like": 12,
  "title": "GET isteğini anlamak",
  "body": "GET metodu genellikle sunucudan veri okumak veya listelemek için kullanılır."
},
```

```
[
  {
    "postId": 1,
    "id": 1,
    "name": "Açıklayıcı bir başlangıç",
    "email": "ogrenci1@example.com",
    "body": "API kavramını kısa ve anlaşılır biçimde açıklıyor."
  },
  {
    "postId": 1,
    "id": 2,
    "name": "Güzel örnek",
    "email": "ogrenci2@example.com",
    "body": "İstemci ve sunucu örneği de eklenebilir."
  },
  {
    "postId": 2,
    "id": 3,
    "name": "Request ve response",
    "email": "ogrenci3@example.com",
    "body": "İstek ve yanıt mantığını anlamak HTTP için önemli."
  },
  {
    "postId": 3,
    "id": 4,
    "name": "GET denemesi",
    "email": "ogrenci4@example.com",
    "body": "Tarayıcı üzerinden GET isteğini kolayca test ettim."
  },
  {
    "postId": 4,
    "id": 5,
    "name": "POST denemesi",
    "email": "ogrenci5@example.com",
    "body": "JSON gövdesi göndererek yeni kayıt yanıtı aldım."
  },
]
```

Örnek

<https://furkandurmus.com/tools/api/posts>

Ve

<https://furkandurmus.com/tools/api/comments>

adreslerindeki verileri localde depolayarak ilgili postun yorumlarını tek datada toplayınız. Data yazdırıldığında yandaki gibi çıktı üretmesi beklenmektedir.

```
{
  "post_id": 1,
  "post_baslik": "API nedir?",
  "post_icerik": "API, farklı yazılım bileşenlerinin belirli kurallar üzerinden haberleşmesini sağlayan bir arayüzdür.",
  "post_yorumlar": [
    {
      "postId": 1,
      "id": 1,
      "name": "Açıklayıcı bir başlangıç",
      "email": "ogrenci1@example.com",
      "body": "API kavramını kısa ve anlaşılır biçimde açıklıyor."
    },
    {
      "postId": 1,
      "id": 2,
      "name": "Güzel örnek",
      "email": "ogrenci2@example.com",
      "body": "İstemci ve sunucu örneği de eklenebilir."
    }
  ]
},
{
  "post_id": 2,
  "post_baslik": "HTTP ile ilk adım",
  "post_icerik": "İstemci sunucuya bir istek gönderir, sunucu da isteği işleyerek uygun bir yanıt döndürür.",
  "post_yorumlar": [
    {
      "postId": 2,
      "id": 3,
      "name": "Request ve response",
      "email": "ogrenci3@example.com",
      "body": "İstek ve yanıt mantığını anlamak HTTP için önemli."
    }
  ]
},
{
  "post_id": 3,
  "post_baslik": "GET isteğini anlamak",
  "post_icerik": "GET metodu genellikle sunucudan veri okumak veya listelemek için kullanılır.",
  "post_yorumlar": [
    {
      "postId": 3,
      "id": 4,
      "name": "GET denemesi",
      "email": "ogrenci4@example.com",
      "body": "Tarayıcı üzerinden GET isteğini kolayca test ettim."
    }
  ]
}
```

Örnek

<https://furkandurmus.com/tools/api/posts>

Ve

<https://furkandurmus.com/tools/api/comments>

adreslerindeki verileri localde depolayarak ilgili postun yorumlarını tek datada toplayınız. Data yazdırıldığında yandaki gibi çıktı üretmesi beklenmektedir.

```
import requests

url = "https://furkandurmus.com/tools/api/posts"

request_1 = requests.get(url)
posts = request_1.json()

request_2 = requests.get("https://furkandurmus.com/tools/api/comments")
comments = request_2.json()

data = []

for post in posts:
    post_id = post["id"]
    post_baslik = post["title"]
    post_icerik = post["body"]
    post_yorumlar = []

    for comment in comments:
        if comment["postId"] == post_id:
            post_yorumlar.append(comment)

    data.append({
        "post_id": post_id,
        "post_baslik": post_baslik,
        "post_icerik": post_icerik,
        "post_yorumlar": post_yorumlar
    })

print(data)
```



Tesekkürler



www.furkandurmus.com

