

## Python'da kendi DNS sistemimizi oluşturma

### Adımlar:

1. UDP Soketi ile DNS İsteklerini Dinleme
2. Gelen İstekleri Okuma ve DNS Paketini Çözme
3. Önceden Tanımlı Yanıtlarla Cevap Verme veya Gerçek DNS Sunucularına Sorgu Gönderme
4. Yanıtı İstemciye Gönderme

\*\*Aşağıda basit bir python DNS örneği verilmiştir.

Gerekli olan kütüphanelerin install kodu
------------------------------------------

pip install dnslib
--------------------

\*\*Uygulamanın kaynak kodu aşağıda verilmiştir.

DnsExample.py
---------------

```
import socket
from dnslib import DNSRecord, RR, A

# Önceden belirlenmiş DNS kayıtları
DNS_KAYITLARI = {
    "example.com": "192.168.1.100",
    "test.com": "192.168.1.101",
    "google.com": "8.8.8.8",
}

# DNS Sunucusu Ayarları
HOST = "0.0.0.0" # Tüm ağ arayüzlerinde dinle
PORT = 80 # DNS'in varsayılan portu

# UDP soketi oluştur
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((HOST, PORT))

print(f"[*] Basit DNS sunucusu {PORT} portunda çalışıyor...")

while True:
    # Gelen DNS sorgusunu dinle
    data, addr = sock.recvfrom(512) # Maksimum 512 byte'lık veri alabiliriz

    # DNS isteğini çözümü
    dns_request = DNSRecord.parse(data)
    domain_name = str(dns_request.q.qname).strip(".")

    print(f"[+] Sorgu alındı: {domain_name} ({addr})")

    # Yanıt oluştur
    dns_response = DNSRecord(dns_request.header)
    dns_response.add_question(dns_request.q)
    if domain_name in DNS_KAYITLARI:
        ip_address = DNS_KAYITLARI[domain_name]
        dns_response.add_answer(RR(domain_name, rdata=A(ip_address)))
        print(f"[*] {domain_name} için {ip_address} adresi döndürüldü.")
    else:
        print(f"[-] {domain_name} için kayıt bulunamadı.")
    # Yanıtı istemciye gönder
    sock.sendto(dns_response.pack(), addr)
```

**\*\*Uygulamayı test etme**

nslookup example.com 127.0.0.1:80	Test kodu (Cmd ve komut ekranına yazılacak)
Server: 127.0.0.1 Address: 127.0.0.1#53  Name: example.com Address: 192.168.1.100	Çıktı sonuçları Not: port yukarıda verilen kodda 53 ile deneyin çalışmaz ise 80 portunu deneyin.

## Zookeeper ile dağıtık senkronizasyon

### Etcd ve Zookeeper nedir ?

etcd ve Apache Zookeeper, dağıtık sistemlerde yapılandırma yönetimi, hizmet keşfi ve koordinasyon için kullanılan popüler dağıtık anahtar-değer mağazalarıdır. Bu araçlar, yüksek erişilebilirlik (high availability) ve tutarlılık (consistency) sağlayarak mikro servisler ve büyük ölçekli sistemler için kritik bir rol oynar.

### Docker Kullanarak etcd Kurulumu:

Docker üzerinden etcd kurulumu yapmak için altta verilen kodu kullanın
<pre>docker run -d --name etcd -p 2379:2379 quay.io/coreos/etcd:v3.5.9 etcd --advertise-client-urls http://0.0.0.0:2379 --listen-client-urls http://0.0.0.0:2379</pre>

**\*\* Docker üzerinden kurulum sağlandıktan sonra**

<pre>docker exec etcd env ETCDCTL_API=3</pre>
-----------------------------------------------

Kodu ile API versiyonunuzu belirliyorsunuz ve alttaki komutlar ile etcd ye veri ekleyebilirsiniz

<pre>docker exec etcd etcdctl put /config/db_url "mysql://localhost:3306" (veri eklendi sonuç olarak "OK" dönecek)</pre>
<pre>docker exec etcd etcdctl get /config/db_url (veri okundu sonuç olarak "/config/db_url mysql://localhost:3306" dönecek)</pre>

**\*\* Yine etcd' ni hem Windows hemde diğer işletim sisitemleri için docker kullanmadan farklı kurulum senaryosu bulunmakta aşağıda manuel kurulum adımları bulunmaktadır.**

1. İlk önce verilen adresten <https://github.com/etcd-io/etcd/releases> sizin işletiminize ait olan etcd sürümünü indirin.
2. Ardından etcd dosya dizinine CMD komut istemcisinden girin ve etcd klasörüne içine girin ardından 

```
"etcd --advertise-client-urls http://127.0.0.1:2379 --listen-client-urls http://127.0.0.1:2379"
```

 komutunu çalıştırın
3. Son olarak ise çalışır durumda ise yukarıda verilen veri ekleme kodunu ve okuma kodunu çalıştırarak veri ekleyip okuyabilirsiniz. (Kurulum sağlandıktan sora denilen kod kısmından başlayın)

### Docker üzerinden Zookeeper kurulumu:

Zookeeper indirme ve kulum kodu
docker pull zookeeper (Eğer belirli bir sürüm indirecekseniz “zookeeper:3.8” kodunu kullanın)
Zookeeper çalıştırma kodu
docker run -d --name zookeeper -p 2181:2181 zookeeper
Zookeeper CLI ekranına bağlanma kodu
docker exec -it zookeeper zkCli.sh
Veri ekleme
create /config "" (Yeni bir node oluşturulur fakat içi boş bir node)
create /config/db_url "mysql://localhost:3306" (oluşturulan node doldurma kodu)
get /config/db_url (Oluşturulan node verisini alma)
Yaml dosya düzeni
version: '3' services: zookeeper: image: zookeeper container_name: zookeeper ports: - "2181:2181" restart: always