





neden bu hata

17 saniye için düşündüm >

Bu hata Python'un DNS üzerinden `api.collectapi.com` adresini IP'ye çevirmeye çalışırken başarısız olduğunu gösteriyor (getaddrinfo failed). Yani istek aslında API'den dönmeden önce, ağ katmanında takılıyor.

Başlıca sebepler şunlar olabilir:

#### 1. İnternet bağlantısı yok

- Çalışma ortamınız (PC, sanal makine, Docker konteyneri vs.) internete çıkamıyor olabilir.
- Tarayıcıdan veya terminalden `ping api.collectapi.com` veya `nslookup api.collectapi.com` komutlarıyla testi deneyin.

#### 2. DNS veya ağ ayarları sorunlu

- DNS sunucu ayarlarınız kaldırılmış ya da yanlış.
- Bir kurumsal ağ/şirket VPN'i içindeyseniz dış DNS sorguları engelleniyor olabilir.

#### 3. Firewall veya proxy engeli

- Windows / Linux güvenlik duvarı `api.collectapi.com` 'a çıkışı bloke ediyor olabilir.
- Eğer bir proxy kullanmanız gerekiyorsa `HTTP_PROXY` / `HTTPS_PROXY` ortam değişkenlerini ayarlamanız gerekebilir.

#### 4. Yanlış URL yazımı

- Kodunuzda `BASE_URL = "https://api.collectapi.com/news/getNews"` yazdığınızdan emin olun — başında mutlaka `https://` olsun.

### Çözüm önerileri

1. İnternet bağlantınızı kontrol edin; tarayıcıdan `https://api.collectapi.com/news/getNews?country=tr&tag=general` adresine erişebiliyor musunuz?
2. Terminalde `nslookup api.collectapi.com` ile DNS sorgusu yapın.
3. Gerekliyse VPN'i / proxy ayarlarını düzenleyin veya geçici olarak kapatın.
4. Kodunuzun bulunduğu ortamda (örneğin Docker) `~x` erişiminin etkin olduğundan emin olun.

favori sistemi, arama kutusu ve dışa aktarma implemente etmek istiyorum

```
import tkinter as tk
from tkinter import ttk, messagebox
import sqlite3
import requests
import webbrowser
import os
from dotenv import load_dotenv

load_dotenv()

API_KEY = os.getenv('COLLECTAPI_KEY')
BASE_URL = "https://api.collectapi.com/news/getNews"
VALID_TAGS = ("general", "sport", "economy", "technology")
COUNTRIES = ("tr", "de", "en", "ru")
news_data = []
```

Aşağıdaki üç geliştirmeyi adım adım uygulayarak projenizi daha işlevsel hâle getirebilirsiniz:

## ✓ 1. Favori Sistemi

a. Veritabanı tablosuna `is_favorite` sütunu ekleyin:

```
python

c.execute("ALTER TABLE articles ADD COLUMN is_favorite INTEGER DEFAULT 0")
# Bu kod sadece ilk kez çalıştırıldığında hata verir, o yüzden try-except'e alabilirsiniz.
```

b. Haber detayına "★ Favorilere Ekle" butonu:

`open_detail` fonksiyonuna aşağıdaki kodu ekleyin:

```
python

def add_to_favorites():
    c.execute("UPDATE articles SET is_favorite=1 WHERE url=?", (url,))
    conn.commit()
    messagebox.showinfo("Favori", "Haber favorilere eklendi.")

btn_fav = ttk.Button(win, text="★ Favorilere Ekle", command=add_to_favorites)
btn_fav.pack(pady=5)
```

c. Sadece favorileri gösteren bir buton ekleyin:

Üst menüye:



## 2. Arama Kutusu

a. Üst menüye arama çubuğu ekleyin:

```
python

entry_search = ttk.Entry(frame_top, width=20)
entry_search.pack(side="left", padx=(5, 0))
entry_search.insert(0, "Ara...")

btn_search = ttk.Button(frame_top, text="🔍", command=lambda: search_articles(entry_search.get))
btn_search.pack(side="left", padx=(5, 0))
```

b. Arama fonksiyonu:

```
python

def search_articles(keyword):
    tree.delete(*tree.get_children())
    news_data.clear()

    keyword = f"%{keyword}%"
    c.execute("""
        SELECT title, source, substr(published_at,1,10), news_id, description, url
        FROM articles
        WHERE (title LIKE ? OR description LIKE ?)
        ORDER BY rowid DESC
    """, (keyword, keyword))

    rows = c.fetchall()
    for title, source, date, news_id, desc, url in rows:
        c.execute("SELECT image FROM articles WHERE news_id=?", (news_id,))
        img_result = c.fetchone()
        img_url = img_result[0] if img_result else ""
        tree.insert("", "end", values=(title[:120], source))
        news_data.append((title, desc, url, img_url))
```

## 3. Dışa Aktarma (Export CSV)

a. Menüye buton ekleyin:

```
python

btn_export = ttk.Button(frame_top, text="📄 Dışa Aktar", command=lambda: export_to_csv(news_data))
btn_export.pack(side="left", padx=(5, 0))
```

b. Export fonksiyonu:



```
lst_news.bind("<Double-Button-1>", open_detail)
```

```
load_articles("tr", "general")  
root.mainloop()
```

haber detaylarında resimlerin gözükmesini istiyorum

Harika bir istek! 🍌 Haber detay penceresinde **haber görsellerini** göstermek istiyorsan, şu adımları uygulayarak çözebiliriz:

## 🎯 Hedef

Her haberin detay penceresinde:

- ✅ Başlık
- ✅ Açıklama
- ✅ 🖱️ Tarayıcıda aç butonu
- ✅ 🖼️ Ve şimdi **resim** de gözüksün.

## ✅ Gerekli Değişiklikler

Tkinter içinde **resim göstermek** için şu adımlar gerekir:

1. `PIL` (Pillow) ile internetten görseli çekmek ve göstermek
2. `news_data` içinde `image` URL'lerini de saklamak
3. `open_detail()` fonksiyonuna resmi dahil etmek

## 1. Pillow Kurulu mu?

Eğer yüklü değilse, terminalden bir kez çalıştırmam yeterli:

```
bash
```

[📋 Copy](#)[✎ Edit](#)

```
pip install pillow
```

## 2. `news_data` Listesini Güncelle (resim URL'si dahil)

```
...data ...
```

[📋 Copy](#)[✎ Edit](#)

Uploaded an image

özellikle bu haber kutusunu daha iyi hale getirmek istiyorum

Görselden anladığım kadarıyla, haber listesi ( `Listbox` ) şu an:

- Satırların hizalaması bozuk görünüyor (özellikle kaynak adı sağda kırılıyor),
- Yazılar taşarak okunması zor hale geliyor,
- Görsel estetikten uzak (tek font, düz renk, hizasız içerik).

## Hedefimiz

`Listbox` yerine `Treeview` veya `Text widget` kullanarak:

- **Başlık** ve **kaynağı** ayrı sütunlarda göstermek
- Satır hizalarını korumak
- Daha okunaklı ve düzenli bir görünüm sağlamak

## ✓ Çözüm: `ttk.Treeview` ile Çok Sütunlu Liste

Aşağıdaki örnekte `Listbox` yerine `Treeview` kullandık. Böylece:

- Her haberin başlığı ve kaynağı ayrı sütunlara yerleştirildi.
- Genişlikler ayarlanabilir.
- Scrollbar uyumlu çalışıyor.

## 🔧 Yeni `Treeview` ile Arayüz Kod Parçası

Aşağıdaki kod parçası sadece liste kısmını günceller:

```
python

# Liste Alanı
frame_list = ttk.Frame(root)
frame_list.pack(fill="both", expand=True, padx=10, pady=10)

columns = ("title", "source")
tree = ttk.Treeview(frame_list, columns=columns, show="headings")
tree.heading("title", text="Haber Başlığı")
tree.heading("source", text="Kaynak")
```

Copy

Edit