



**Bigdata Teknoloji**

# Django API Test Projesi

Bigdata Teknoloji Backend Developer Aday Projesi

# İçindekiler

---

|   |   |
|---|---|
| Giriş.....                                  | 3 |
| Proje İstekleri: .....                      | 3 |
| Projenin Teslimi.....                       | 4 |
| Veri Çekilecek Dış API bilgileri.....       | 5 |
| Lokasyon Bilgisi Sorgu API: LocationIQ..... | 5 |
| Kullanıcı bilgileri: .....                  | 5 |
| Sorgu URL: .....                            | 5 |
| Hava Tahmin Sorgu API: DarkSKY .....        | 5 |
| Kullanıcı bilgileri: .....                  | 5 |
| Sorgu URL: .....                            | 5 |
| Opsiyonel Algoritma Sorusu: .....           | 6 |

# Giriş

---

Ekibimiz için aradığımız backend developer pozisyonu için prosedürel olarak aşağıdaki projenin aday tarafından tamamlanmasını bekliyoruz.

Buradaki amaç adayın soruna yaklaşımı, kod yazım tarzını daha iyi anlayabilmek. Söz konusu proje, içeriğinden de anlaşılabilir gibi, bizim herhangi bir ürünümüzde kullanılmayacaktır, sadece adayı tanımak için tanımlanmış bir projedir.

Proje detaylarımız aşağıdaki gibidir, herhangi bir sorunuz olması durumunda bizimle irtibata geçebilirsiniz.

## Proje İstekleri:

---

1. Web API olarak gerçekleşmesini beklediğimiz uygulama temel olarak kullanıcı tarafından gönderilen istekteki konum ismine (örneğin İstanbul) göre:
  - Mevcut sıcaklığı
  - Günün kalanı içinde beklenen en düşük ve en yüksek değerleri
  - Önümüzdeki bir hafta içinde beklenen en düşük ve en yüksek sıcaklığıkullanıcıya JSON formatında dönmeli.
2. Tüm proje (Django API servisi, database, ve caching katmanı) Docker ve docker-compose ile çalışmalı.
3. API endpointleri django rest framework kullanılarak yazılmalı.
4. API dokümantasyonu için Swagger kullanılmalı ve API çağrıları oradan yapılabilmelidir.
5. Projenin mobil ve web uygulama olarak son kullanıcılara sunulması durumunda, aşağıdaki iki faz için gerekli olabilecek sistem mimarisini çizerek gösterin (herhangi bir çizim aracı kullanılabilir veya kağıt üzerine elle çizim yapıp görüntüsünü paylaşabilirsiniz)
  - a. Faz 1: 100 kullanıcı
  - b. Faz 2: 10.000.000 kullanıcı

**Opsiyonel:** Basit bir kullanıcı giriş sistemi (user authorization) ile bu API çağrıları korunmalı. Kullanıcı sadece giriş yaptıktan sonra bu çağrılardan cevap alabilmeli.

## Akış

---

Verilen konum ismine ait latitude longitude bilgileri locationiq APIsinden alınmalı (dönülen ilk lat, lon bilgisi kullanılabilir).

latitude longitude bilgileriyle darksky APIsinden bugünkü ve o hafta beklenen en düşük ve en yüksek sıcaklık alınıp kullanıcıya dönülmeli.

Bu bilgiler ilişkisel bir veritabanında tutulmalı

Gelen istekteki konum bilgisine ait bugün yapılmış sorgu varsa dış API çağrısı yapılmadan, önceden kaydedilmiş değerler veritabanından alınıp kullanıcıya gönderilmeli.

Bu bilgiler aynı zamanda bir caching katmanında da tutulmalı, caching katmanında en fazla 50 ve son 1 saat içinde yapılmış sorgular ve cevapları yer almalı. Cachete yer alan sorgular için veritabanına gidilmeden doğrudan cacheteki yanıt istemciye dönülmeli.

## Projenin Teslimi

---

Projenizi özel veya açık bir git repository üzerinden bizimle paylaşabilirsiniz.

# Veri Çekilecek Dış API bilgileri

---

Projede ihtiyaç duyacağınız verileri çekebilmek için ücretsiz API servisleri ve bunlarda kullanabileceğiniz hesaplar aşağıda verilmiştir. Eğer aşağıdaki hesaplar üzerinden erişim sağlayamazsanız ücretsiz olarak yeni hesap açabilirsiniz.

## Lokasyon Bilgisi Sorgu API: LocationIQ

Kullanıcı bilgileri:

`https://my.locationiq.com`

Kullanıcı: `d2152063@urhen.com`

Token: `a1779b7817b3b2`

Sorgu URL:

`https://eu1.locationiq.com/v1/search.php?key=a1779b7817b3b2&q=[location]&format=json`

## Hava Tahmin Sorgu API: DarkSKY

Kullanıcı bilgileri:

`https://darksky.net/dev/account`

Kullanıcı: `d2152063@urhen.com`

Şifre: `123456ab`

Key: `f3146e0fc78b4930d41a60703c08e2ae`

Sorgu URL:

`https://api.darksky.net/forecast/[key]/[latitude],[longitude]`

# Opsiyonel Algoritma Sorusu:

---

Yukardaki proje haricinde algoritma tasarım yetkinliğinizi göstermek için aşağı verilen problemi çözen bir python fonksiyonu yazınız.

**Problem:** Büyük bir daire içindeki  $2n$  sayıdaki kişi el sıkışmak için  $n$  çift oluşturur. El sıkışacak çiftlerin elleri birbiri ile çaprazlamadan (yani üst üste gelmeden) kaç şekilde eşleşip tokalaşabilirler?  $n$  sayısının 0'dan 5000'e kadar alabileceği tüm değerler için gruptaki insanların tokalaşabileceği yolların sayısını bulan bir python fonksiyonu yazın.

İlave notlar:

- Her kişinin sabit bir konumu vardır
- Her kişi arasındaki mesafe önemli değildir.
- $f(n)$  gruptaki insanların tokalaşabileceği yolların sayısını ifade eder.
- $f(0) = f(1) = 1$

Algoritma yapısı ve hesaplama süresi değerlendirmede dikkate alınacaktır. İhtiyaç duymanız durumunda algoritma hakkında lütfen sorularınızı iletin.