

Patika Bitirme Projesi

<https://openweathermap.org/api> kullanarak Şehir Adı ve Tarih Aralığı bilgilerine göre hava kirliliği bilgisinin okunması ve veritabanına kaydedilmesi sürecinin yürütüleceği bir Restfull uygulamanın Spring Boot Framework kullanılarak yazılması ve isteğe bağlı önyüzünün yazılması.

Gereksinimler:

- Geliştirilecek olan Restfull servis public bir API olan <https://openweathermap.org/api> aracılığıyla gerekli bilgileri alacaktır. Bu API'de yer alan Historical Air Pollution (<https://openweathermap.org/api/air-pollution#history>) ve Geocoding (<https://openweathermap.org/api/geocoding-api>) endpointleri kullanılacaktır.
- Şehir Adı bilgisi ile Geocoding api yardımıyla ilgili şehrin koordinat bilgileri alınabilir.
- Sorgulanan hava kirliliği bilgisi, ilgili endpoint'in döndüğü kirli madde bilgilerinden sadece **Carbon monoxide (CO)**, **Ozone (O3)**, **Sulphur dioxide (SO2)** bilgilerini içerecektir.
- Okunan kirletici madde konsantrasyonu bilgileri https://en.wikipedia.org/wiki/Air_quality_index#CAQI sitesinde yer alan "**AQI Category, Pollutants and Health Breakpoints**" tablosu kullanılarak "**Good, Satisfactory, Moderate, Poor, Severe ve Hazardous**" olarak sınıflandırılacaktır.
- Geliştirilen serviste **Şehir Adı** ve **Tarih Aralığı** bilgileri ile tarihsel olarak hava kirliliği bilgisi sorgulanabilecek, veri tabanında daha önceden var olan kayıtlar silinebilecektir. Detaylar şu şekildedir:
 - Sorgulama yaparken Şehir Adı **zorunlu** olacaktır, Tarih Aralığı ise girilmediği takdirde **varsayılan** olarak son 1 hafta olarak alınacaktır.
 - Sadece **London, Barcelona, Ankara, Tokyo ve Mumbai** şehirleri için sorgulama yapılacaktır.
 - Tarihsel veri erişimi API tarafından **27 Kasım 2020'den günümüze** sınırlandırıldığından bu tarihler dışındaki girdiler için tarih aralığının yanlışlığı ile alakalı hata mesajı döndürülecektir.
 - Sorgulama sonucu olarak **Tarihe** göre sıralanmış, sınıflandırılmış kirli madde bilgileri JSON formatında dönülecektir.
 - **Örnek istek:**
<endpoint_address>?city=ankara&startDate=16-05-2021&endDate=17-05-2021

■ **Cevap:**

```
{ "City": "Ankara", "Results": [
  { "Date": "16-05-2021", "Categories": [ { "CO": "Satisfactory" }, { "SO2": "Moderate" }, { "O3": "Satisfactory" } ] },
  { "Date": "17-05-2021", "Categories": [ { "CO": "Good" }, { "SO2": "Severe" }, { "O3": "Satisfactory" } ] } ] }
```

- Her sorgu isteğinde öncelikle veritabanına bakılacak, sadece veritabanında olmayan tarihler için API sorgusu atılacaktır. API'den getirilen sonuçlar veritabanına kaydedilecektir. Getirilen sonuçlar için veritabanından mı API'den mi getirildiği bilgisi loglanacaktır. Örnek:
 1. Sorgu: **Paris, 05-01-2022 ile 10-01-2022** -> veritabanında olmadığından bilgiler tarih aralığı ile API'den alınacak ve veritabanına kaydedilecek.
 2. Sorgu: **Ankara, 05-01-2022 ile 10-01-2022** -> veritabanında olmadığından bilgiler tarih aralığı ile API'den alınacak ve veritabanına kaydedilecek.
 3. Sorgu: **Paris, 01-01-2022 ile 15-01-2022** ->
 - a. 01-01-2022 ile 04-01-2022 API'den alınacak veritabanına kaydedilecek.
 - b. 05-01-2022 ile 10-01-2022 veritabanından alınacak.
 - c. 11-01-2022 ile 15-01-2022 API'den alınacak veritabanına kaydedilecek.
- Yapılacak tüm istekler için her zaman doğru bilgi girileceği varsayılmıştır, şehir adı ve tarih formatı gibi bilgilerin doğruluğunun kontrol edilmesine gerek yoktur.

Projenin Beklenenleri:

- Backend projesinin belirtilen kurallara göre doğru çalışır olması (readme dosyasının eklenmesi ya da Dockerfile yazılması) (20 puan)
- Kodun kalitesine (Clean Code), yapılandırılmasına (Structure) dikkat edilmesi, yeni özellikler için geliştirmeye açık olması ve anlaşılabilir olması. (SOLID principles) (15 puan)
- Mikroservis mimarisi ve pratiklerine (Distributed Systems), teknolojiye ve bu alandaki yeniliklere bakış açınızı yansıtır olması. (15 puan)
- Test yazılması (15 puan)
- Design Patterns kullanımı (10 puan)
- Dokümantasyon (Swagger, OpenApi vb) (5 puan)
- NoSQL, RDBMS (ORM) ve Hibernate (JPA) gibi teknolojilerin kullanımı (10 puan)
- Loglama mekanizmasının kurulması (10 puan)