

Uçuş Arama Servisi

Java, Spring Boot, Maven, SOAP ve REST ile text dosyasındaki bilgileri okuyarak arama kriteri ve arama sorgusu ile çalışan uçuş arama web servisi, IntelliJ geliştirme ortamında hazırlandı. SOAP API, SOAP UI ile; REST API, Postman ile test edildi. Proje structure'ı:

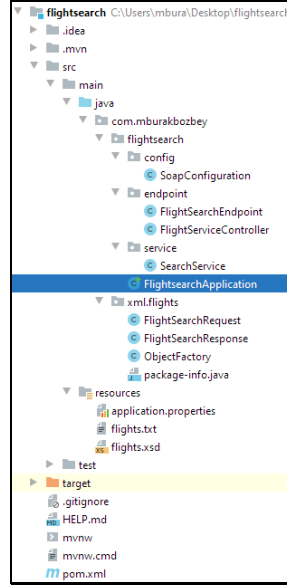


Fig. 1 – Proje Yapısı

1. config > **SoapConfiguration.java** – SOAP API için gereken konfigürasyon
2. endpoint > **FlightSearchEndpoint.java** – SOAP API için gereken Endpoint
3. endpoint > **FlightServiceController.java** – REST API için gereken RestController
4. **FlightsearchApplication.java** – Çalıştırılması gereken Application
5. service > **SearchService.java** – txt Dosyasının okunduğu, sorgunun txt üzerinde aratılıp geri döndürüldüğü arama servisi
6. resources > **flights.xsd** – Girdi tiplerinin belirtildiği .xsd dosyası
7. resources > **flights.txt** – Uçuş bilgilerinin tutulduğu, “;” ile ayrılmış text dosyası.

1. Projenin Çalıştırılması

Proje'nin çalıştırılabilmesi için maven ile dependencyler yüklendikten sonra, FlightsearchApplication.java run edilmelidir. SOAP için wsdl, <http://localhost:8080/ws/searchServiceAPI.wsdl> üzerinde tanımlanmıştır. Arama sorgusu için **SOAP UI** üzerinde yeni SOAP projesi açılmalı, belirtilen wsdl url'si ile oluşturulduktan sonra hazır olarak gelen xml format içerisinde test edilebilir. Örnek XML girdisi 5.kısımda belirtilmiştir. REST API için, <http://localhost:8080/search> request url'ine JSON formatında **Postman** ile test edilebilir. Örnek JSON girdisi 5.kısımda belirtilmiştir.

2. Spring Boot & Maven

2.1 Pom.xml Dosyasının Oluşturulması

Spring boot kullanmak için gerekli olan proje tanımı, java versiyonu, dependencyler ve pluginler pom.xml dosyası içerisine eklendi.

2.1 Xsd Dosyasının Oluşturulması

Belirtilen girdi ve çıktılar XML formatında flights.xsd dosyası içerisinde tanımlandı. FlightSearchRequest elementi altında searchCriteria integer olarak, searchKey ise string olarak tanımlandı. FlightSearchResponse elementi altında aramaya uygun olan çıktılar string list olarak tanımlandı. Xsd dosyası Figür 2'de belirtilmiştir.

```

<?xml version='1.0'?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.mburakbozbey.com/xml/flights"
  targetNamespace="http://www.mburakbozbey.com/xml/flights"
  elementFormDefault="qualified">

  <xs:element name="FlightSearchRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="searchCriteria" type="xs:int"/>
        <xs:element name="searchKey" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="FlightSearchResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="FlightList" block="extension"
          minOccurs="0" maxOccurs="unbounded" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>

```

Fig. 2 – flights.xsd

Girdi: searchCriteria(int), searchKey(string)

- searchCriteria sırasıyla firma, model, kalkış yeri, varış yeri ve saat sütunlarının numerik olarak kodlanması halinde integer olarak tanımlandı.
[firma, model, kalkış yeri, varış yeri, saat] → [0, 1, 2, 3, 4]

Çıktı: FlightList

- Sorgu sonucunda uygun olan uçuş bilgileri, string listesi halinde, delimiter olarak "---" kullanılarak geri döndürülmektedir.

3. Arama Servisi

(src/main/java/com.mburakbozbey/flightsearch/service/SearchService.java)

Girdilerin mevcut spesifikasyonların dışında olması durumunda gerekli hata mesajları, Assertion olarak her bir case için belirlendi. Girdilerin hatalı olması durumunda Figür 3’deki hata mesajları tanımlandı. Bunun dışında verinin bulunduğu text dosyasının okunmaması durumu da exception olarak eklendi.

```

Assert.notNull(searchCriteria, message: "Search criteria must not be null.");
Assert.isTrue( expression: searchCriteria < 5 && searchCriteria >= 1, message: "Incorrect search criteria, must be in [0, 1, 2, 3, 4].");
Assert.hasLength(searchKey, message: "Search key must not be null and must not the empty.");

```

Fig. 3 – Girdiler için tanımlanan hata mesajları

Uygun uçuşun bulunmaması durumunda Figür 4’teki hata mesajı tanımlandı.

```

Assert.notEmpty(check.getFlightList(), message: "No flight is available for the search criteria and key.");

```

Fig. 4 – Çıktı için tanımlanan hata mesajı

Uçuş bilgileri verisi, Türkçe karakterler için utf-8 encoding ile beraber Scanner ile okunduktan sonra, verilen searchCriteria’ya göre searchKey’in lowercase haliyle veri içerisinde karşılık gelen elemanların da lowercase’e dönüştürülmüş haliyle karşılaştırılıp listeye eklendi. Birden fazla uygun uçuş bilgisi olması durumunda, "---" delimiter ekstra olarak eklendi. Elde edilen nihai liste FlightSearchResponse sınıfında List<String> flightList şeklinde yer almaktadır.

4. SOAP API

SOAP için konfigürasyon (src/main/java/com.mburakbozbey/flightsearch/config/SoapConfiguration.java) ve endpoint sınıfı (src/main/java/com.mburakbozbey/flightsearch/endpoint/FlightSearchEndpoint.java) oluşturuldu. SOAP UI üzerinden uygulamanın çıktıları kontrol edildi ve bir önceki bölümde belirtilen hata mesajları test edildi. Çeşitli senaryolar, dokümanın 6.1. bölümünde belirtilmiştir. Örnek XML requesti:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:flig="http://www.mburakbozbey.com/xml/flights">
  <soapenv:Header/>
  <soapenv:Body>
    <flig:FlightSearchRequest>
      <flig:searchCriteria>0</flig:searchCriteria>
      <flig:searchKey>thy</flig:searchKey>
    </flig:FlightSearchRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

5. REST API

REST API için controller java sınıfı (src\main\java\com\mburakbozbey\flightsearch\endpoint\FlightServiceController.java) oluşturuldu. @RestController altında hem XML, hem de JSON consume ve produce edecek şekilde @PostMapping tanımlandı. Gönderilen sorgu <http://localhost:8080/search> request URL'ine gidecek şekilde tanımlandı.

Dokümantasyonun Arama Servisi kısmında belirtilen hata mesajları, Postman üzerinde farklı JSON sorgularıyla test edildi. Error handling, spring boot Assertion'ları ile yapıldı. Her bir Assertion için hata mesajında status code 500, "error" ise Internal Server Error şeklinde gelirken, uygun hata mesajı "message" kısmında yer almaktadır. Çeşitli senaryolar, dokümanın 6.2 bölümünde belirtilmiştir. Örnek JSON requesti:

```
{
  "searchCriteria" : 0,
  "searchKey" : "thy"
}
```

6. API Test

Farklı girdiler için response'lar ve hatalı girdiler için hata mesajları, SOAP UI ve Postman üzerinde test edilmiştir. Uygun olarak tanımlanan searchCriteria ve searchKey için doğru sonuçlar elde edildi. Ekran görüntüleri bölüm 6.1 ve 6.2'ye eklenmiştir.

6.1 SOAP API & SOAP UI

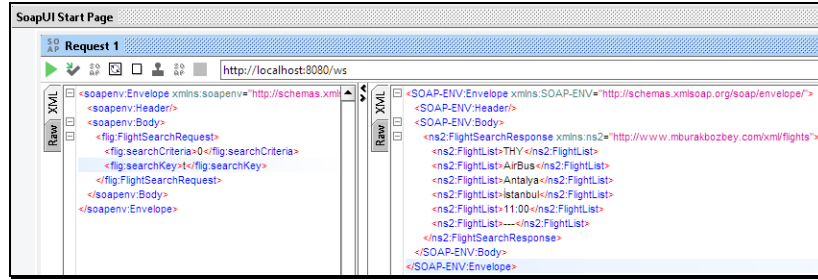


Fig.5 – searchCriteria: 0, searchKey: t ve XML response

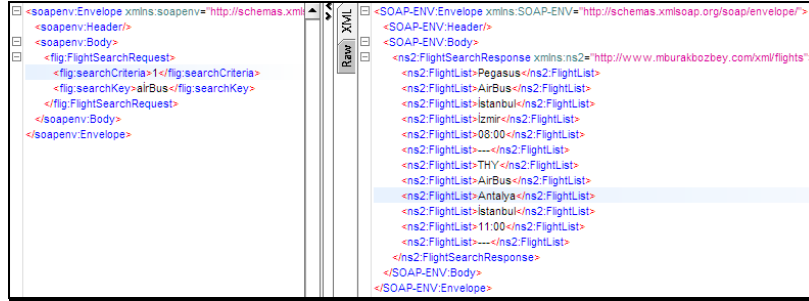


Fig.6 – searchCriteria: 1, searchKey: airBus ve XML response

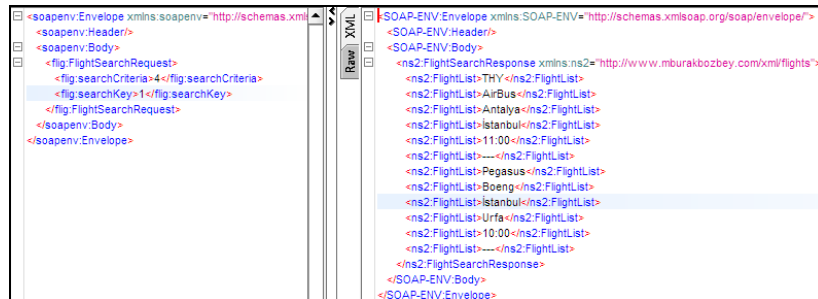


Fig.7 – searchCriteria: 4, searchKey: 1 ve XML response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <flig:FlightSearchRequest>
      <flig:searchCriteria>3</flig:searchCriteria>
      <flig:searchKey>İS</flig:searchKey>
    </flig:FlightSearchRequest>
  </soapenv:Body>
</soapenv:Envelope>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:FlightSearchResponse xmlns:ns2="http://www.mburakbozbey.com/xml/flights">
      <ns2:FlightList>THY</ns2:FlightList>
      <ns2:FlightList>AirBus</ns2:FlightList>
      <ns2:FlightList>Antalya</ns2:FlightList>
      <ns2:FlightList>İstanbul</ns2:FlightList>
      <ns2:FlightList>11:00</ns2:FlightList>
      <ns2:FlightList>....</ns2:FlightList>
    </ns2:FlightSearchResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fig. 8 – searchCriteria: 3, searchKey: İS ve XML response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <flig:FlightSearchRequest>
      <flig:searchCriteria>3</flig:searchCriteria>
      <flig:searchKey>Ankara</flig:searchKey>
    </flig:FlightSearchRequest>
  </soapenv:Body>
</soapenv:Envelope>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring xml:lang="en">No flight is available for the search criteria and key.</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fig. 9 – searchCriteria: 3, searchKey: Ankara ve Uçuş Bulunamadı Hata Mesajı

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <flig:FlightSearchRequest>
      <flig:searchCriteria>22</flig:searchCriteria>
      <flig:searchKey>İS</flig:searchKey>
    </flig:FlightSearchRequest>
  </soapenv:Body>
</soapenv:Envelope>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring xml:lang="en">Incorrect search criteria, must be in [0, 1, 2, 3, 4]</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fig. 10 – searchCriteria: 3, searchKey: Ankara ve Hatalı searchCriteria Hata Mesajı

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <flig:FlightSearchRequest>
      <flig:searchCriteria>0</flig:searchCriteria>
      <flig:searchKey></flig:searchKey>
    </flig:FlightSearchRequest>
  </soapenv:Body>
</soapenv:Envelope>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring xml:lang="en">Search key must not be null and must not be empty.</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fig. 11 – searchCriteria: 3, searchKey: Ankara ve searchKey Girilmedi Hata Mesajı

6.1 REST API & Postman

```
POST http://localhost:8080/search

{"searchCriteria": 0, "searchKey": "pE"}

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<FlightSearchResponse xmlns="http://www.mburakbozbey.com/xml/flights">
  <FlightList>Pegasus</FlightList>
  <FlightList>AirBus</FlightList>
  <FlightList>İstanbul</FlightList>
  <FlightList>İzmir</FlightList>
  <FlightList>08:00</FlightList>
  <FlightList>...</FlightList>
  <FlightList>Pegasus</FlightList>
  <FlightList>Boeing</FlightList>
  <FlightList>İstanbul</FlightList>
  <FlightList>Urfa</FlightList>
  <FlightList>10:00</FlightList>
  <FlightList>...</FlightList>
</FlightSearchResponse>
```

Fig. 12 – searchCriteria: 0, searchKey: pE ve XML Response

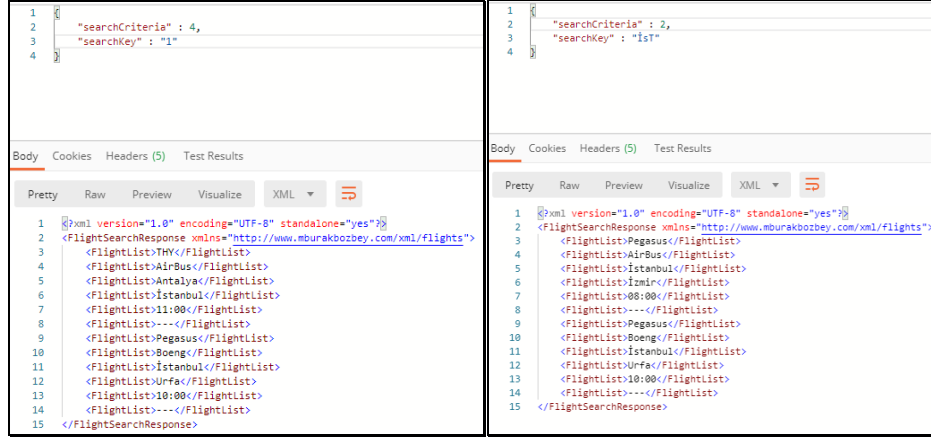


Fig. 13 – searchCriteria: 4, searchKey: 1 ve XML Response | searchCriteria: 2, searchKey: İST ve XML Response

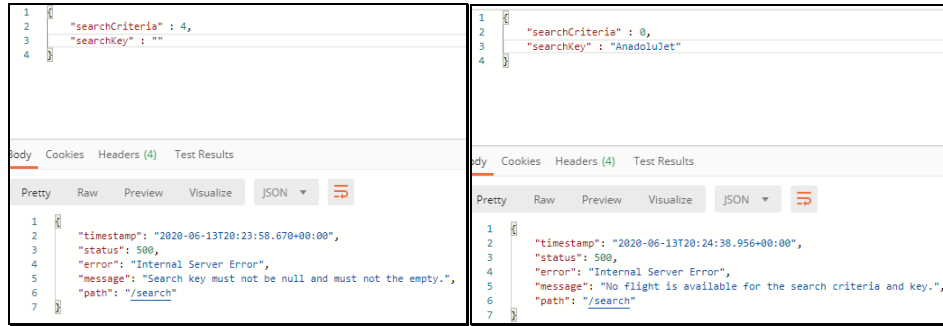


Fig. 14 – searchCriteria: 4, searchKey: "" ve searchKey Girilmedi Hata Mesajı | searchCriteria: 0, searchKey: AnadoluJet ve Uçuş Bulunamadı Hata Mesajı

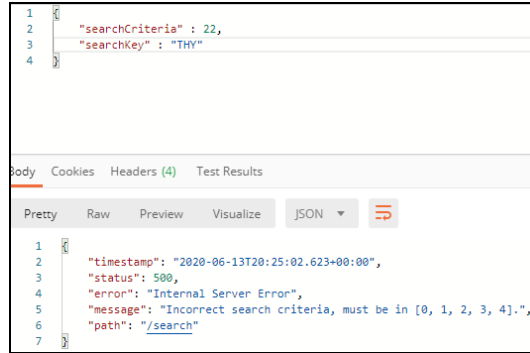


Fig. 15 – searchCriteria: 22, searchKey: THY ve Hatalı searchCriteria Hata Mesajı