



TURKCELL GNCYTNK PROJESİ

Mevlude Nur ÇAKIROĞLU

1. Spring Boot <https://start.spring.io/> adresinden starter projesi oluşturup IntelliJ Editörünü kullanarak işlemine başladım
2. Bana Mail Olarak Gönderilmiş txt Dosyasını istenilen yapıya uygun olacak bir Package oluşturdum içerisinde class'larımı tanımladım. Bu class benim veri kaynağıdır.
3. Verilerimi ve Kullanıcı Erişimi için bir Package oluşturdum Service adında service package benim verilerimi dosyadan okuyup class'a aktarmak için kullanacağı ve istek yapıldığında buradan değerlerimi geri döndüreceğim alandır.
4. Service.class da List<Arac> olarak bilgilerimi tutmaktayım; Bir istek yapıldığında DosyaOku() metodumu çağırarak verilerimi Listenin içerisine aktarıyorum.
5. **@RequestMapping ve @GetMapping** Kullanarak Http'den gelen path değerine göre istemcimi yönlendiriyorum
6. Http'den gelen isteginde bulunan koşula göre verileri mi filtreden geçirip response ediyorum

DosyaOku() Fonksiyonu

```
protected void DosyaOku() throws IOException {
    CarsList = new ArrayList<>();
    ClassLoader loader = Service.class.getClassLoader();
    FileReader fileReader = new FileReader(ResourceUtils.getFile(this.getClass().getResource("degerler.txt")));
    String line;

    BufferedReader br = new BufferedReader(fileReader);

    while ((line = br.readLine()) != null) {
        String[] deger = line.split(" ");
        Entity arac = new Entity();
        arac.setMarka(deger[0]);
        arac.setModel(deger[1]);
        arac.setSinif(deger[2]);
        CarsList.add(arac);
    }

    br.close();
}
```

Araç Markasına Göre Filtreleme

```
@GetMapping("/marka/{marka}")
@ResponseBody
public List<Entity> searchMarka(@PathVariable("marka") String marka, Model el) throws IOException {
    DosyaOku();
    return CarsList.stream().filter(x -> x.getMarka().toLowerCase().equals(marka.toLowerCase())).collect(Collectors.toList());
}
```

Araç Modeline Göre Filtreleme

```
@GetMapping("/model/{model}")
@ResponseBody
public List<Entity> searchModel(@PathVariable("model") String model, Model el) throws IOException {
    DosyaOku();
    return CarsList.stream().filter(x -> x.getModel().toLowerCase().equals(model.toLowerCase())).collect(Collectors.toList());
}
```

Araç Sınıfına Göre Filtreleme

```
@GetMapping("/sinif/{sinif}")
@ResponseBody
public List<Entity> searchSinif(@PathVariable("sinif") String sinif, Model el) throws IOException {
    DosyaOku();
    return CarsList.stream().filter(x -> x.getSinif().toLowerCase().equals(sinif.toLowerCase())).collect(Collectors.toList());
}
```

Araçların Tümünü Listeleme

```
@RequestMapping(value = "hepsi", method = RequestMethod.GET)
public List<Entity> getCars() throws IOException {
    DosyaOku();
    return CarsList;
}
```

İçerisindeki Kelimeye Göre Listeleme

```
@GetMapping("/ara/{ara}")
@ResponseBody
public List<Entity> searchArama(@PathVariable("ara") String ara, Model el) throws IOException {
    DosyaOku();
    return CarsList.stream().filter(x -> x.getSinif().toLowerCase().contains(ara.toLowerCase()) ||
        x.getMarka().toLowerCase().contains(ara.toLowerCase()) ||
        x.getModel().toLowerCase().contains(ara.toLowerCase())).collect(Collectors.toList());
}
```

Spring Boot çalıştırmak için main fonksiyonu oluşturdum bu fonksiyon ayrı bir package içerisinde yer almaktadır, class içinde diğer package tanıtip onlarıda derleme sırasında dahil etmesini sağlamış oldum.

```
@SpringBootApplication
@ComponentScan(basePackages = {"com.turkcell.Application.entity"})
@ComponentScan(basePackages = {"com.turkcell.Application|service"})
public class Application {

    public static void main(String[] args) { SpringApplication.run(Application.class, args); }

}
```

Postman Uygulaması ile İstek Yapmak

GET localhost:8080/hepsi

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "model": "A3",
4     "marka": "Audi",
5     "sinif": "Otomobil"
6   },
7   {
8     "model": "A4",
9     "marka": "Audi",
10    "sinif": "Otomobil"
11  },
12  {
13    "model": "A5",
14    "marka": "Audi",
15    "sinif": "Otomobil"
16  }
17 ]
```

Değerlerimiz başarılı bir şekilde postman aracı ile geldi, fakat bir web servisi üzerinden istek yapılacak olursa

Access to XMLHttpRequest at 'http://localhost:8080/hepsi' from origin 'null' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.

böyle bir hata ile karşılaşacağız buna çözüm olarak bu fonksiyonu yazdım ve sorun düzelmiş oldu

```
@Bean
public WebMvcConfigurer configurer(){
    return new WebMvcConfigurer() {
        @Override
        public void addCorsMappings(CorsRegistry registry) {
            registry.addMapping(pathPattern: "/").allowedOrigins("*");
        }
    };
}
```

Test etmek için ve görselleştirme bilmek için küçük bir HTML / JS kodu yazdım
DataTables eklentisi ile ekranda tablo şeklinde yazdırdım bu işlemler tek bir index.html
dosyasında oluşturdum (proje içerisinde web klasöründe yer almaktadır)

Arac Sorgulama Test İşlemi

Hepsi

Ara

Show 5 entries

Marka	Model	Sınıf
Audi	A3	Otomobil
Audi	A4	Otomobil
Audi	A5	Otomobil
Audi	A6	Otomobil
Audi	A7	Otomobil

Showing 1 to 5 of 12 entries

Previous 1 2 3 Next

Arac Sorgulama Test İşlemi

a3

Show 5 entries

Model

Hepsi

Marka

Model

Sınıf

Harf Ara

Marka	Model	Sınıf
Audi	A3	Otomobil

Showing 1 to 1 of 1 entries

Previous 1 Next

büyük /küçük harf duyarlı değildir aksi halde kelime doğru olsa dahi veri boş gelirdi bu
yüzden filtre fonksiyonlarda toLowerCase() kullandım