

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 isteğinde 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( name "/degerler.txt")));
   String line;

BufferedReader br = new BufferedReader(fileReader);

while ((line = br.readLine()) != null) {
   String[] deger = line.split( regex: "");
   Entity anac = new Entity();
   arac.setMarka(deger[0]);
   arac.setMarka(deger[0]);
   arac.setMod(deger[1]);
   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

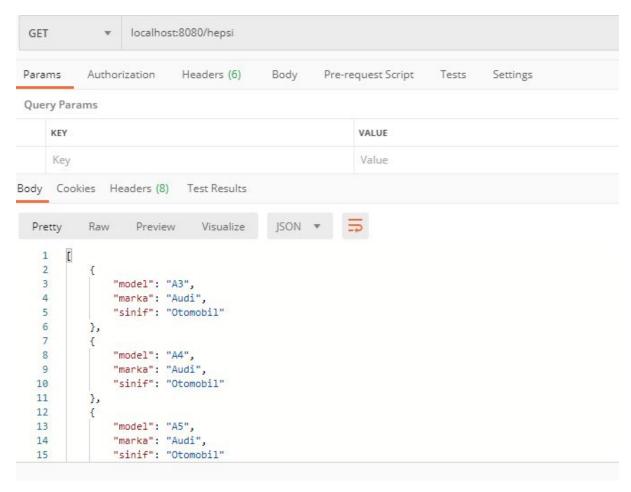
```
QRequestMapping(value = "hepsi", method = RequestMethod.GET)
public List<Entity> getCars() throws IOException {
    DosyaOku();
    return CarsList;
}
```

İçerisindeki Kelimeye Göre Listeleme

Spring Boot çalıştırmak için main fonksiyonu oluşturdum bu fonksiyon ayrı bir package içerisinde yer almaktadır, class içinde diğer package tanıtıp 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



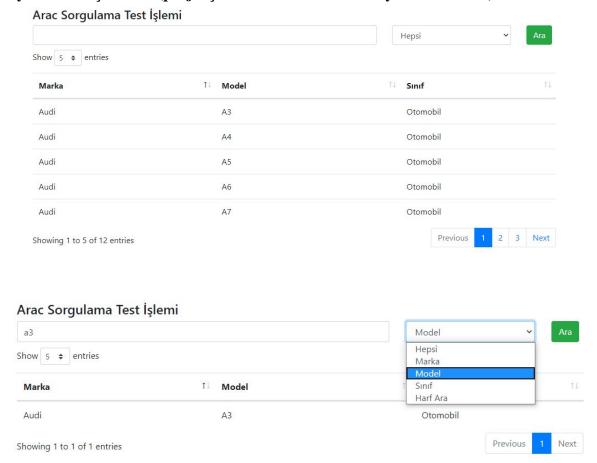
Değerlerimiz başarılı bir şekilde postman aracı ile geldi, fakat bir web servisi üzerinden istek yapılacak olursa

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

```
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)



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