**22.03.2022**

İlgili günde Hibernate bir veritabanında bulunan tablo üzerinde kayıtlı olan tüm verilerin nasıl getirildiği ve bu verinin Front-End(Ön yüz) kısmına nasıl sunulduğu ile ilgili araştırma-çalışma yapılmıştır.

Bir önceki gün yapılan çalışmada, vatandaşlarla ilgili verilerin tutulduğu tablo Hibernate ile mapping edilmişti. Veri getirme ve sunma işlemi bu tablodaki veriler üzerinde gerçekleştirilmiştir. Veri getirme işleminde Service katmanı ile Entity katmanının haberleşmesi gerektiğinden araya Repository katmanı önceki günlerdeki gibi eklenmiştir. Bu katmanda oluşturulan Interface aynı şekilde CrudRepository katmanından kalıtım almaktadır. Diamond operatörünün içerisine Entity katmanında oluşturulmuş olan Citizen sınıfı eklenmiştir (<Citizen,Integer>).

Kullanıcıların veriler için istekte bulunabilmesi ve bu verilerin yanıt olarak geri döndürülebilmesi için Controller katmanı oluşturulmuştur. Bu katman içerisinde CitizenController isminde bir sınıf oluşturulmuştur. Bu sınıf kullanıcı ile etkileşime geçeceği için @RestController anotasyonu ile sınıfa API özelliği kazandırılmıştır. Sınıfın Entity katmanına erişebilmesi için Repository katmanında oluşturulmuş olan CitizenRepository sınıfı ile CitizenController sınıfı @Autowired anotasyonu ile birbirine bağlanmıştır. Kullanıcının verileri talep edebilmesi için getAll() isminde POST isteklerini işleyen bir fonksiyon tanımlanmıştır. Fonksiyon CrudRepository sınıfına ait olan findAll() metodunu çağırarak tablodaki tüm verilere erişmektedir. Veritabanından getirilen nesne verileri ayıklandıktan sonra kullanıcıya Map tipinde bir değişken ile döndürülmektedir. Map veri tipinin kullanılmasındaki amaç, JSON(Javascript Object Notation) syntax’ı ile birebir aynı olmasıdır. Front-End kısmında JSON sözdizimi daha rahat parse edilebilmektedir.

@RestController

@RequestMapping(path = "/user")

public class CitizenController {

@Autowired

private CitizenRepository citizenRepository;

@PostMapping(path = "/getAll")

public ResponseEntity<Object> getAll() {

Iterator<Citizen> citizens=citizenRepository.findAll().iterator();

String[]tmp\_arr;

Map<String, String[]> data = new HashMap<>();

while(citizens.hasNext()){

Citizen user=(Citizen)citizens.next();

tmp\_arr=new String[3];

tmp\_arr[0]=user.getName();

tmp\_arr[1]=user.getSurname();

tmp\_arr[2]=user.getCitizen\_number();

data.put(String.valueOf(user.getId()),tmp\_arr);

}

return new ResponseEntity<>(data, HttpStatus.OK);

}

}

Geliştirilen bu API servisi SoapUI Desktop programı üzerinde test edilmiştir. (EK-8) Kullanıcı ilgili verilere erişmek için POST yöntemiyle <http://localhost:8080/user/getAll> URL adresini çağırması gerekmektedir.