SPRING PROFILE

Spring Profiller, uygulamaları yapılandırmamıza, ayırmamıza ve bunları istediğimiz ortamlarda kullanılabilir hale getirip yönetmemize yardımcı olan yapılardır. Birden fazla ortamlardan kasıt olarak Test, Prod (Production) ve Dev gibi ortamlar örnek gösterilebilir. Bu gibi farklı ortamlarda konfigürasyonlarımız ortamdan ortama farklılık gösterecektir. İşte Spring Profiller, bize doğru ortamlarda doğru konfigürasyonların ayarlanmasında yardımcı olurlar.

Spring Profiller olmadan konfigürasyonları yönetmek zorlu bir süreç olacaktır. Aynı zamanda bu konfigürasyonları birbirleriyle senkron şekilde ayarlamak ve bunların da kontrolünün gerçekleşmesi gerekecektir. Spring profilleri, dağıtım ortamlarıyla sınırlı değildir. Uygulama içerisinde bulunan profildeki herhangi bir seçeneği getirmek için kullanılabilir. Bunun için @Bean kullanılır. Örnek vermek gerecek olursak aşağıda 2 farklı veri tabanı kullanımı için gerekli Spring Profil yazımı ortaya çıkacaktır. Bu sayede 2 farklı veri tabanı için Spring profil oluşturabilirim.

```
@Bean
@Profile("postgre")
public public DatabaseSource postgreDatabaseSource(){
    DatabaseSource databaseSource;

   //İslemler ...

    return databaseSource;

}

@Bean
@Profile("couch")
public public DatabaseSource couchDatabaseSource(){
    DatabaseSource databaseSource;

   //İslemler ....

    return databaseSource;

}
```

Yukarıda oluşturduğumuz gibi Bean oluşturabilirken, istersek XML üzerinden de Bean oluşturma işlemi gerçekleştirebiliriz.

Şimdi sıfırdan örnek bir Spring Boot Profile uygulamasını ayağa kaldıracağız. Bunun için ben Eclipse kullanacağım. Öncelikle yeni bir Spring Boot projesi oluşturuyoruz. İçerisine deneme amaçlı kullanacağım Spring Boot Dev Tools ve Spring Web Dependency özelliklerini ekliyorum.

Sonrasında application.properties dosyasına aşağıdaki eklemeleri yapıyorum. Burada ilk olarak dev kısmı için profil oluşturduğumuz için onu yorum satırı yapmadım. Test ve prod profilleri şuan için yorum şeklinde.

```
spring.application.name = Spring Profile Example

spring.profiles.active=dev
#spring.profiles.active=test
#spring.profiles.active=prod
server.port: 8080
```

Projemize application-dev.yml dosyası ekliyoruz. Burada port bilgisini ve istediğimiz konfigürasyon ayarlarını girebiliriz.

```
spring:
  profiles:
    message:
server:
  port: 8888
```

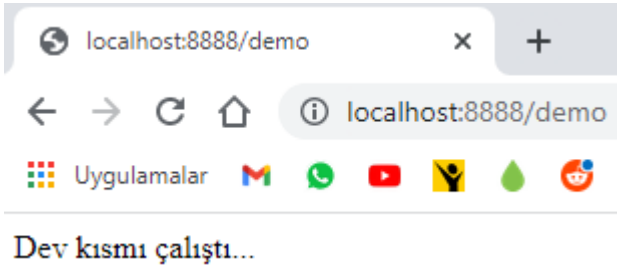
Sonrasında GET ile local tarafta istek atacağımız için DemoController adında bir sınıf oluşturup bu sınıfı @RestController olarak işaretliyorum.

```
public class DemoController {



    @GetMapping(value = "/demo")
    public String message() {
        return "Dev kısmı çalıştı...";

    }

}
```

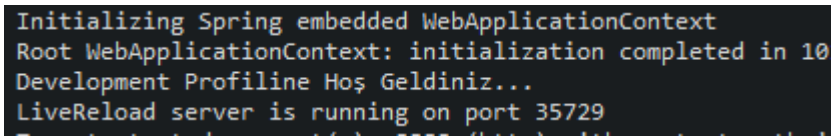
Profillerimi kontrol etmek için bir konfigürasyon sınıfı oluşturuyorum. Burada gerekli @Profile ve @Bean anotasyonlarını kullanarak çıktı elde etmeye çalışacağız.

```
@Configuration
public class ProfileConfiguration {

    private static Logger LOGGER = LoggerFactory.getLogger(ProfileConfiguration.class);

    @Profile(value = "dev")
    @Bean
    public void devconfig() {
        LOGGER.info("Development Profiline Hoş Geldiniz...");
    }

}
```

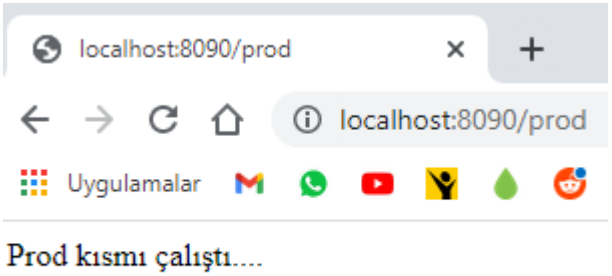
Artık geriye projeyi ayağa kaldırıp sonucu görme zamanı geldi. “Dev” profilimizi 8888 olarak ayarladığımız için localhost:8888/demo olarak local tarafta çalışmayı gözlemleyeceğiz ve çıktı olarak “Dev kısmı çalıştı…” çıktısını bekleyeceğiz.



Bunu LOGGER.info ile birlikte “Development Profiline Hoşgeldiniz” olarak yazdığımız için konsolumuzda bu mesajı görürüz.



Projemize bir tane daha profil ekleyerek daha anlaşılır kılabiliriz. Bunun için “prod” profilini kullanacağım. application.properties dosyasında yorum satırı olarak bulunan “prod” kısmını açmam yeterli olacaktır. Ek olarak application-prod.yml dosyası oluşturup orada bulunan port bilgisini “8090” olarak ayarlacağım. “Prod” için yeni bir Get isteği oluşturup mesaj kısmına “Prod kısmı çalıştı…” göndereceğim.



Bu örneği test vs. için gibi daha çoğaltabiliriz. Profil konusunu anlatmak için bu 2 örneği basit ve sade tutmaya çalıştım. Yine geliştiricilerin işini kolaylaştıran bir yapının kullanımına dair çalışma yapmış olduk. Böylece Profiller oluşturarak, projemizi yönetmemiz kolaylaşacaktır.