

Title

Spring Profile

. . .

@Profile Anotasyonu:

Profile anotasyonu uygulamamızdaki farklı çalışma isterlerine göre programımızın hangi işlevinin çalışacağını çalışacağını isteğimize göre seçmemizi sağlamaktadır.

Bir Bean tanımlamasını nasıl belirli bir Profile’a ait hale getirebileceğimize bakalım. Bunun için @Profile anotasyonunu kullanıyoruz.

Yalnızca development sırasında aktif olması gereken ancak deployment sırasında kullanılmayan bir Bean’e sahip olduğumuzu düşünelim:

Aşağıdaki sınıfı dev profile olarak işaretliyoruz böylece sadece development sırasında mevcut olacak:

```
@Component
@Profile("dev")
public class DevDatasourceConfig
```

Profile adlarının önüne bir NOT operatörü de eklenebilir.

Aşağıdaki örnekte component yalnızca dev profili etkin değilse etkinleştirilir:

```
@Component
@Profile("!dev")
public class DevDatasourceConfig
```

Profile ayrıca XML olarak da yapılandırılabilir:

```
<beans profile="dev">
  <bean id="devDatasourceConfig"
    class="com.batuhankiltac.profiles.DevDatasourceConfig" />
</beans>
```

Spring Profile spring.profiles.active ile belirtilerek Maven Profile aracılığıyla da etkinleştirilebilir:

```
<profiles>
  <profile>
    <id>dev</id>
    <activation>
      <activeByDefault>true</activeByDefault>
    </activation>
    <properties>
      <spring.profiles.active>dev</spring.profiles.active>
    </properties>
  </profile>
  <profile>
    <id>prod</id>
    <properties>
      <spring.profiles.active>prod</spring.profiles.active>
    </properties>
  </profile>
</profiles>
```

. . .

Umarım bu makale yazılım geliştiricilere bir nebze de olsa katkı sağlamıştır.

Başka bir yazıda görüşmek üzere.

İstek ve önerileriniz için bana batuhankiltac@gmail.com adresinden ulaşabilirsiniz.