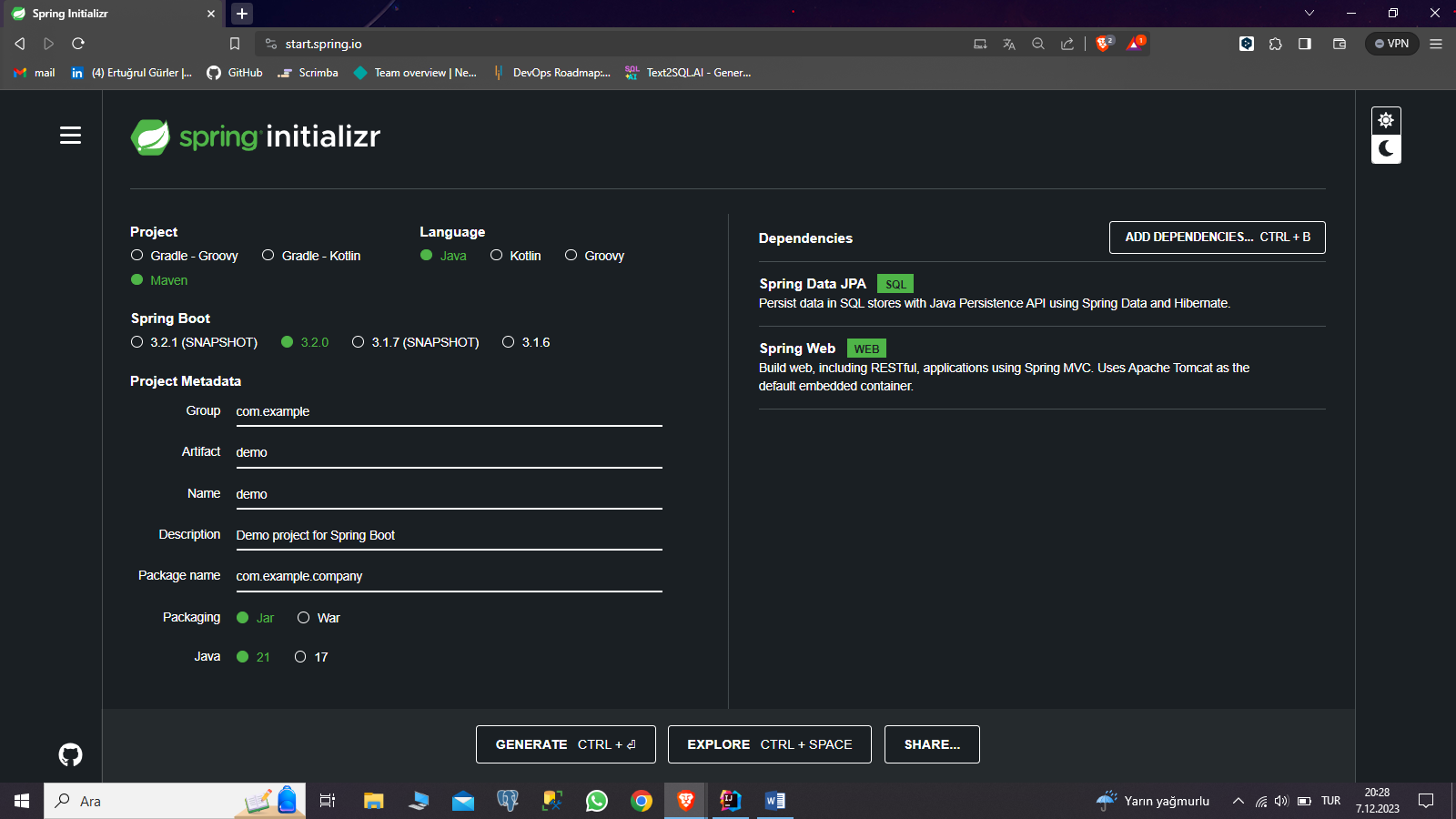
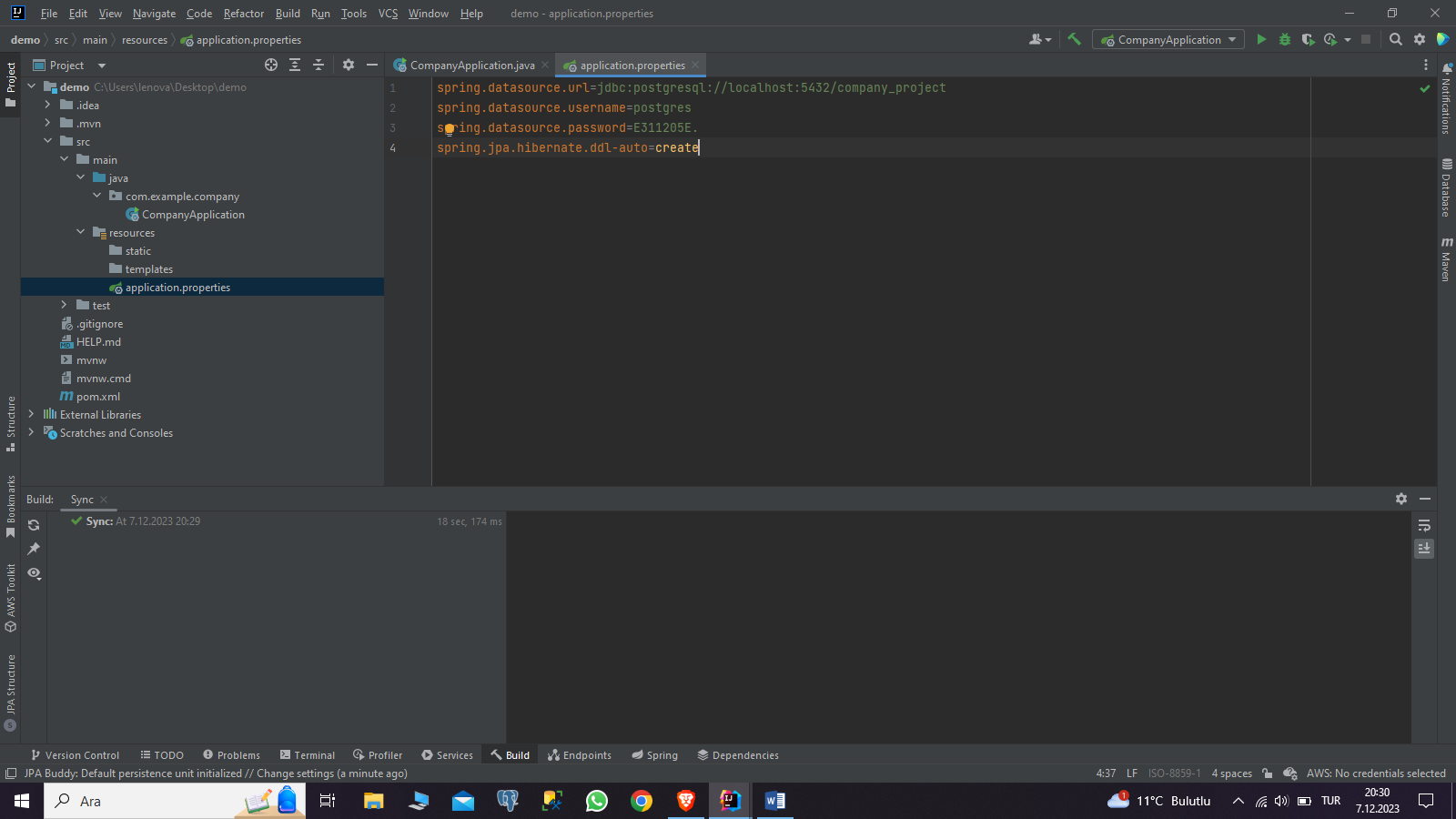
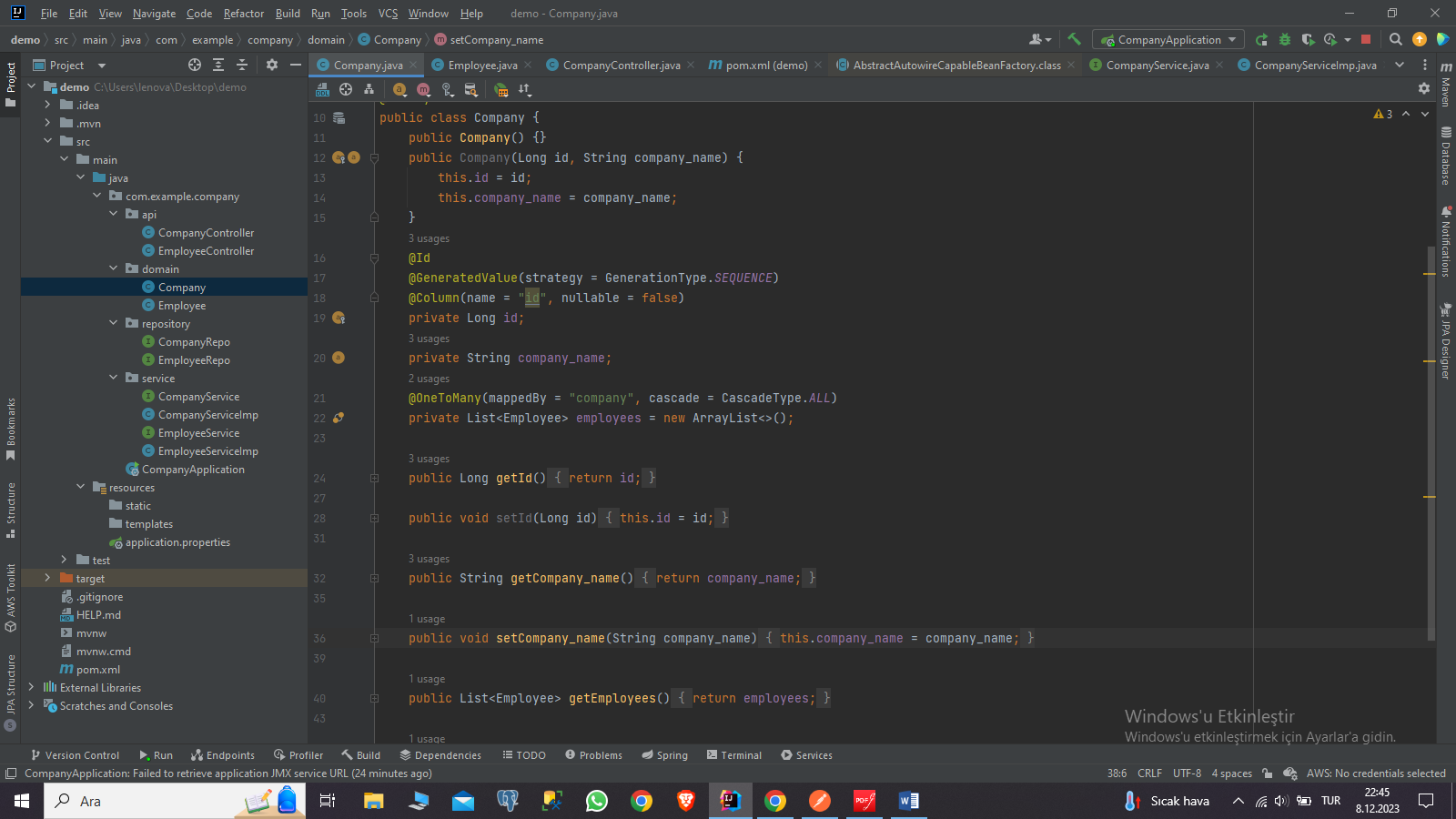
**1-** Spring initializrdan bir springboot projesi alıyoruz. Spring DataJpa veritabanı işemleri için, Spring Web RESTFul servisler kurmak için projeye ekleniyor.



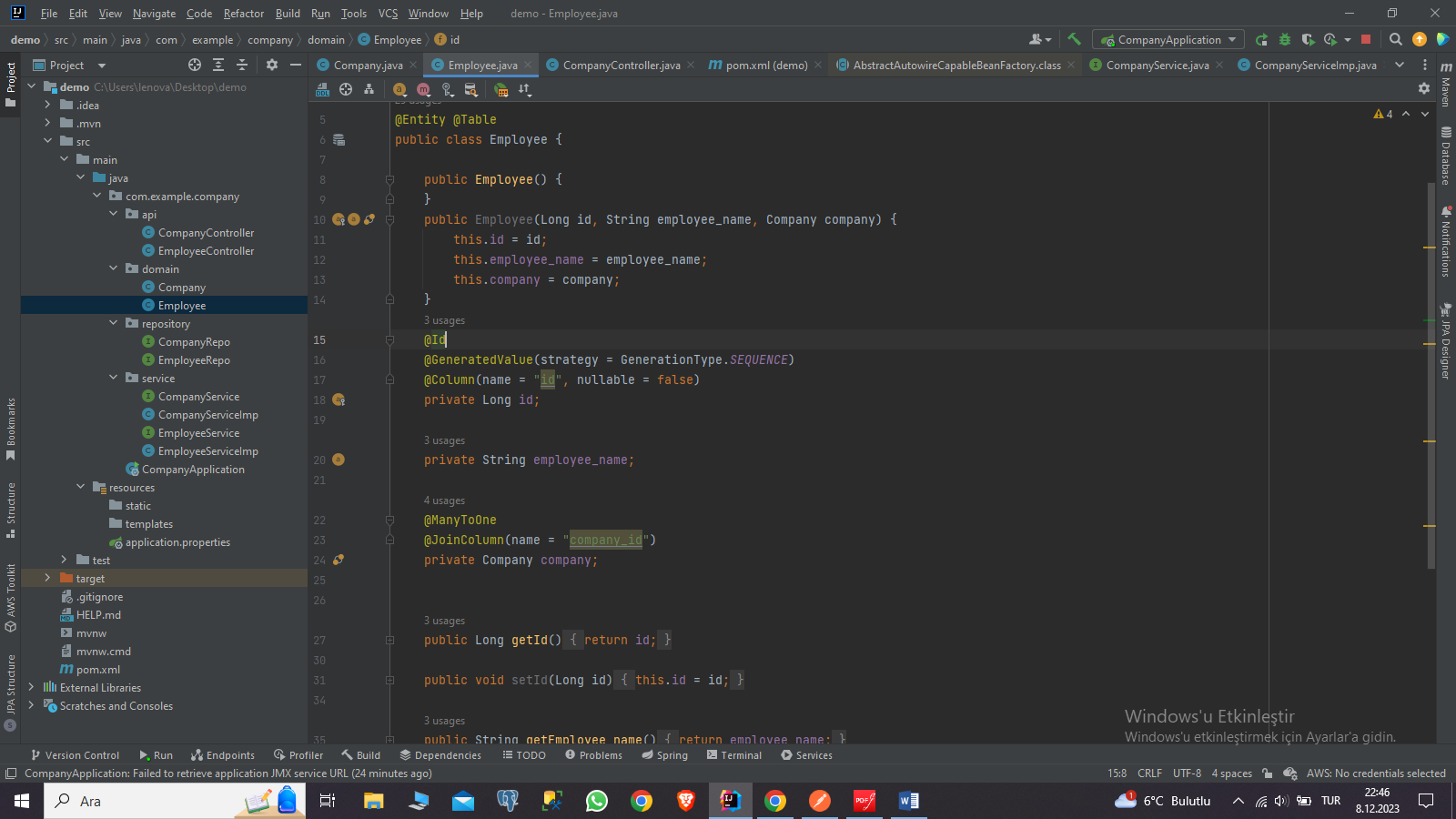
**2-** Local veritabanına bağlanmak için konfigürasyonlar ayarlanıyor.



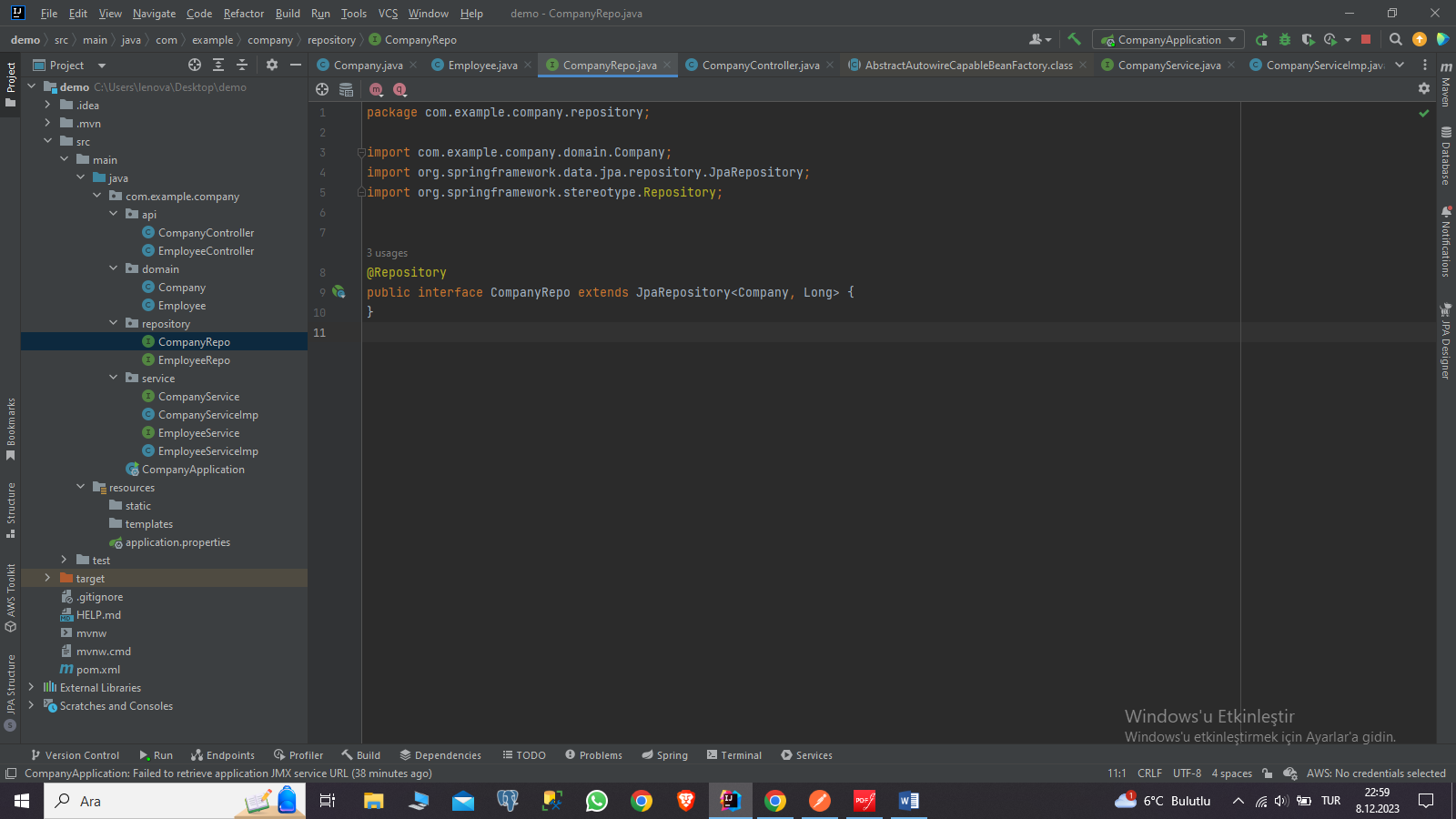
**3-** Company sınıfı oluşturuluyor. Getter ve setterlar yazılıyor.



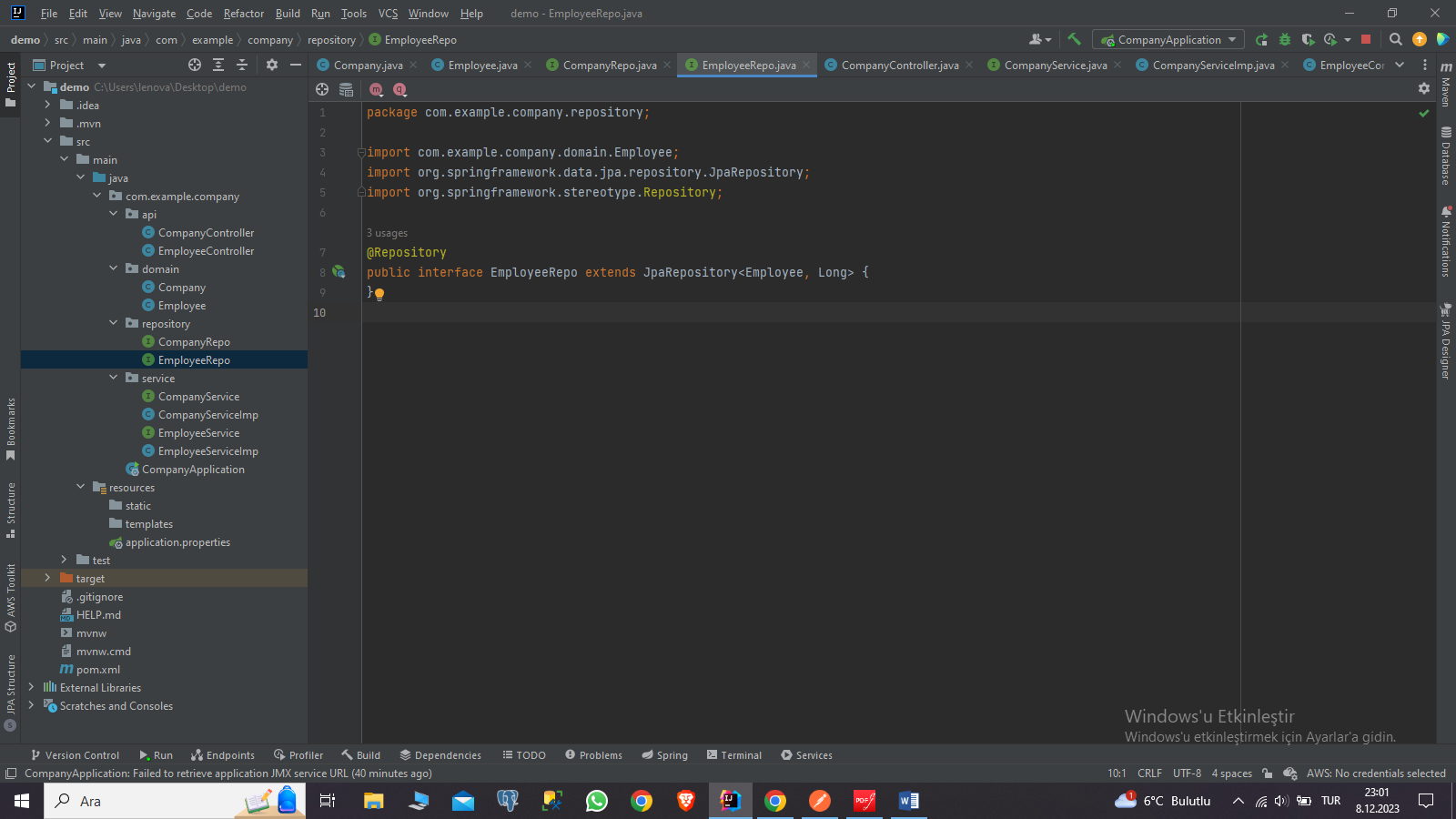
**4-** Employee sınıfı oluşturuluyor. Getter ve setterlar yazılıyor. Hibernate bu classları tablolara dönüştürecek. Classlar, Company ve Employee arasındaki ilişki One to Many – Many to One olacak şekilde ayarlandı.



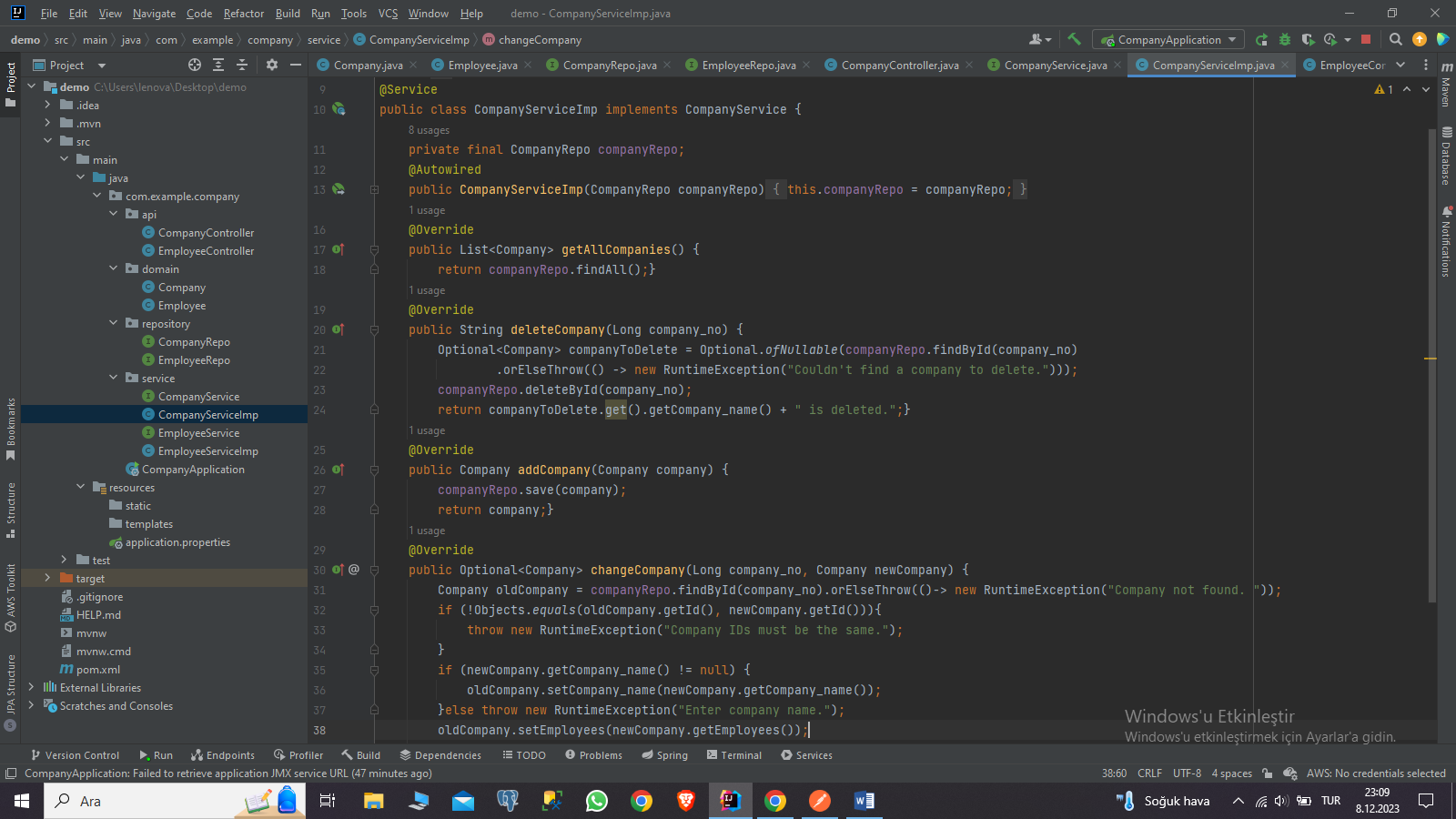
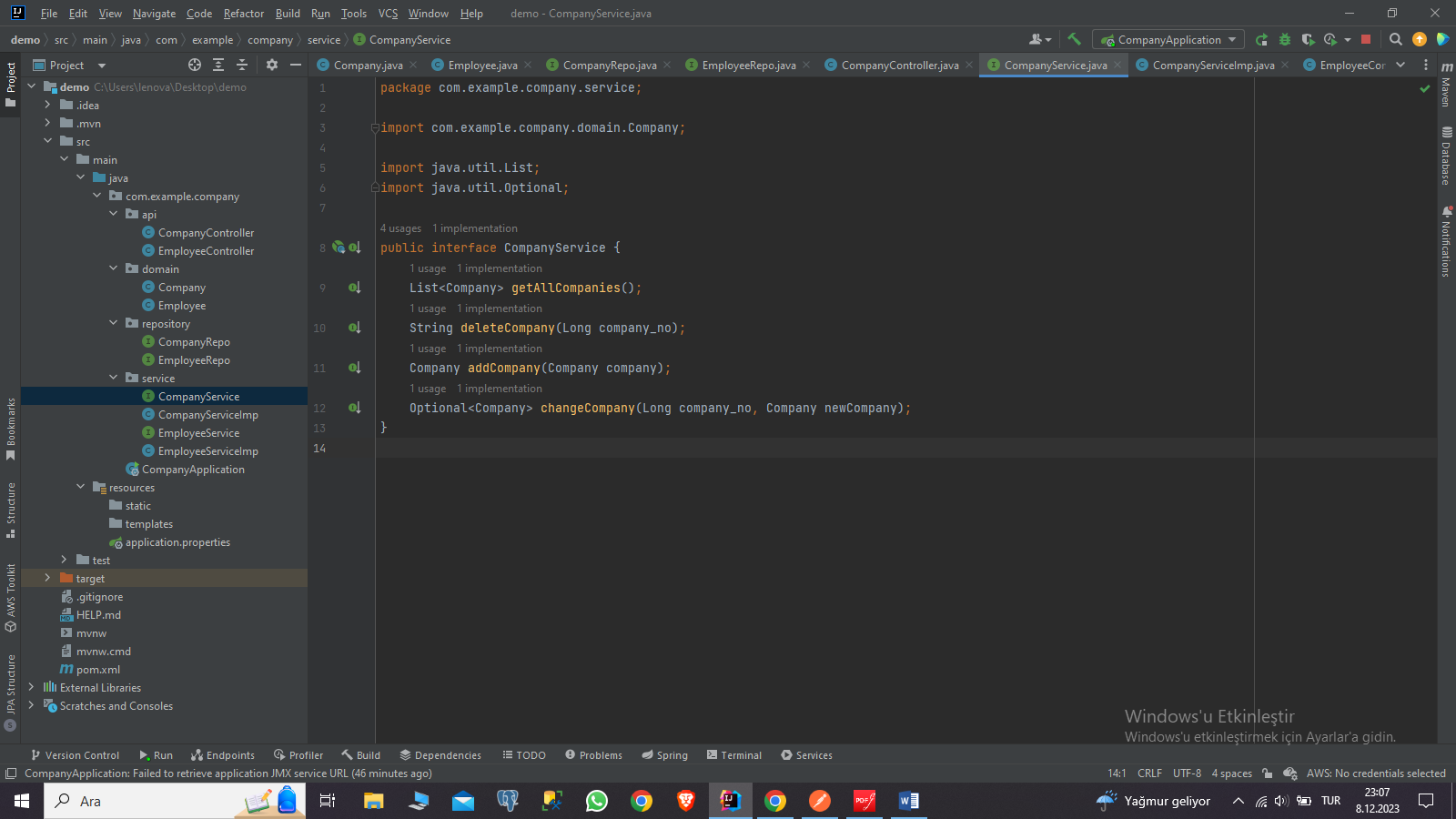
**5-** Şirket içi repository oluşturuldu. JpaRepository verilere ulaşmak ve yönetmek için, gerekli fonksiyonlara ulaşmak için kullanıldı.



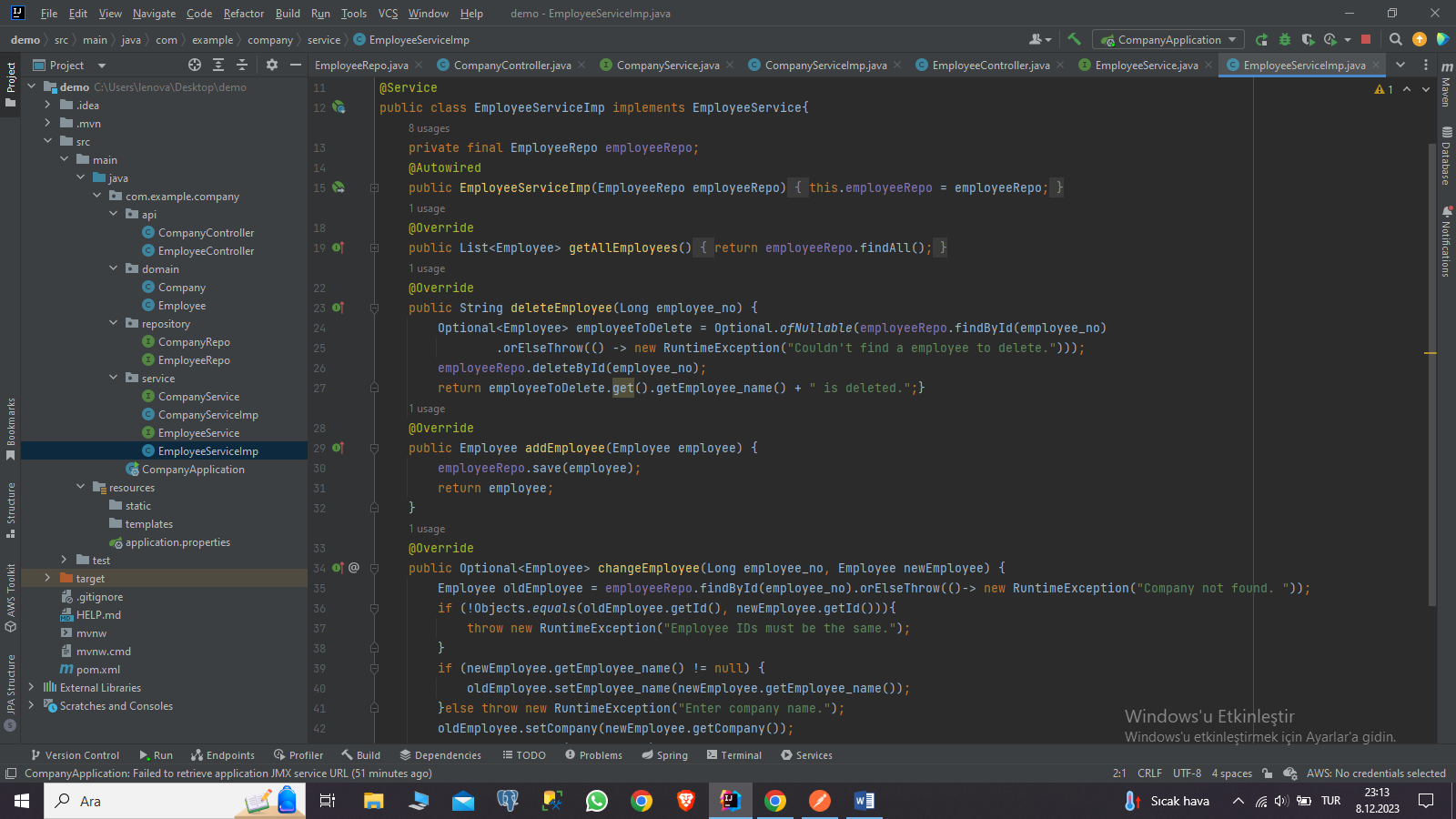
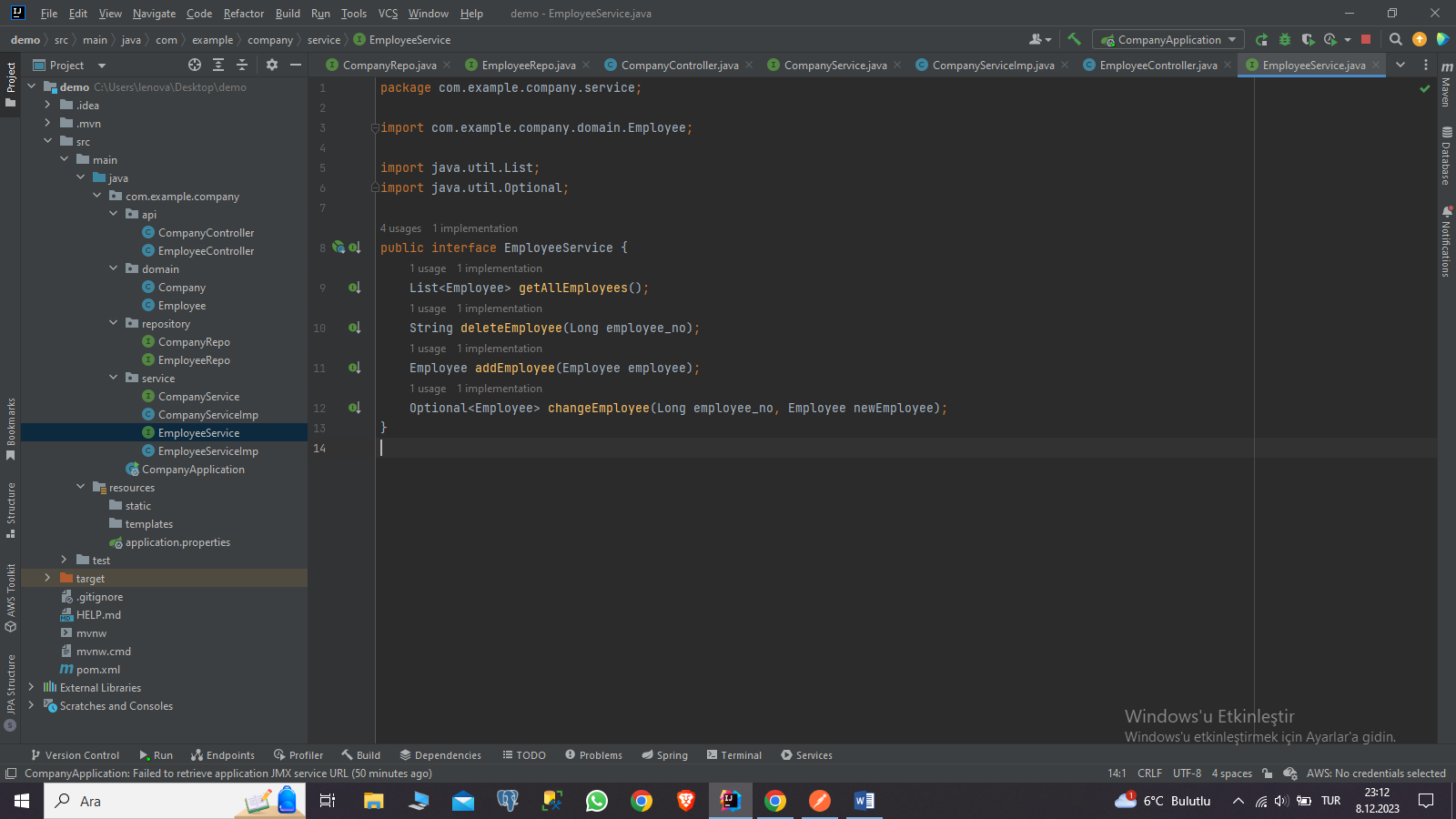
**6-** Employee içi repository oluşturuldu. JpaRepository verilere ulaşmak ve yönetmek için, gerekli fonksiyonlara ulaşmak için kullanıldı.

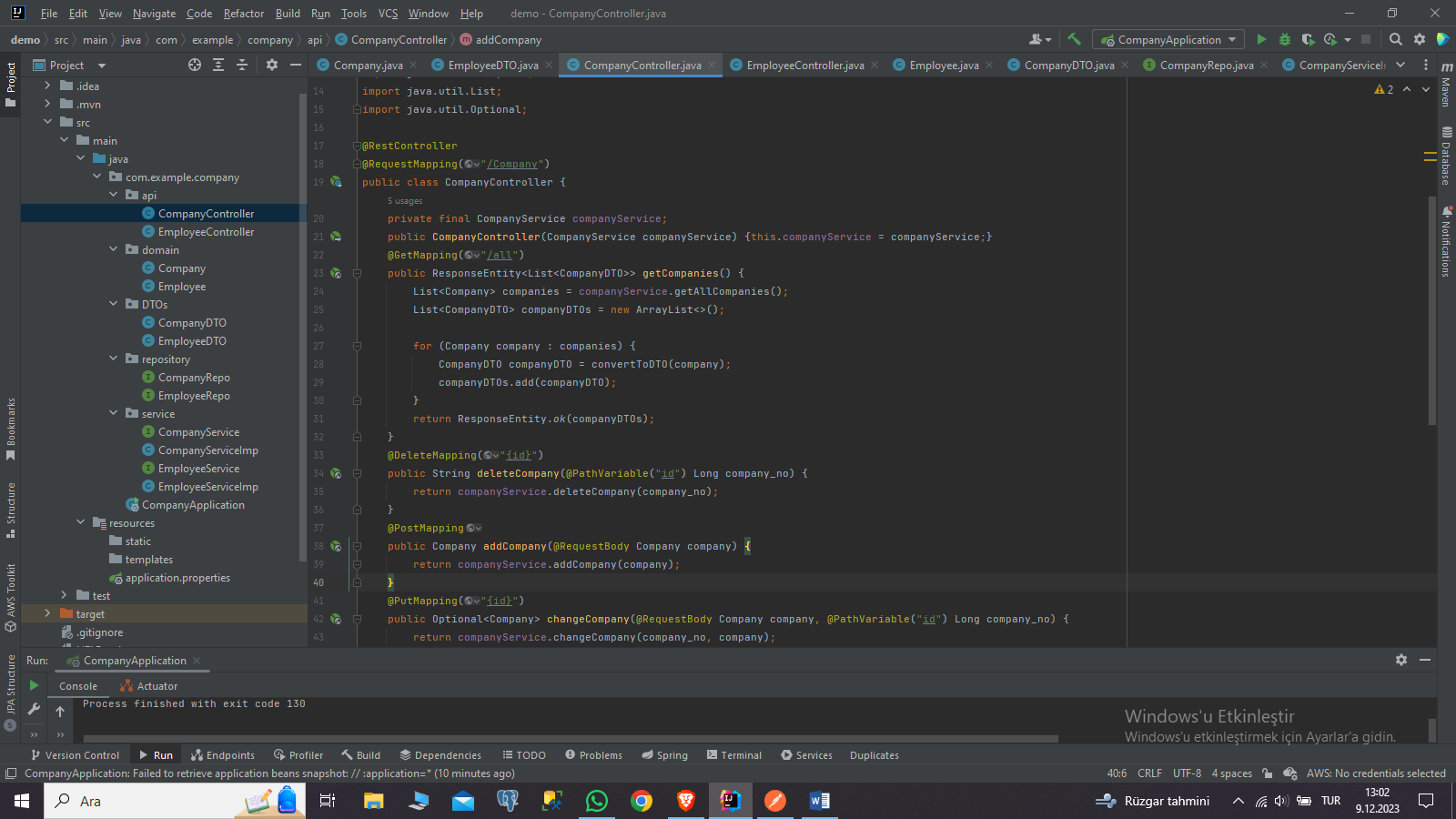


**7-** Esneklik ve bakım kolaylığı için hem servis arayüzü hemde implementasyonu yazıldı. Daha iyi inceleme için kaynak kodları paylaşıldı. Company servisi ve implementasyonu.

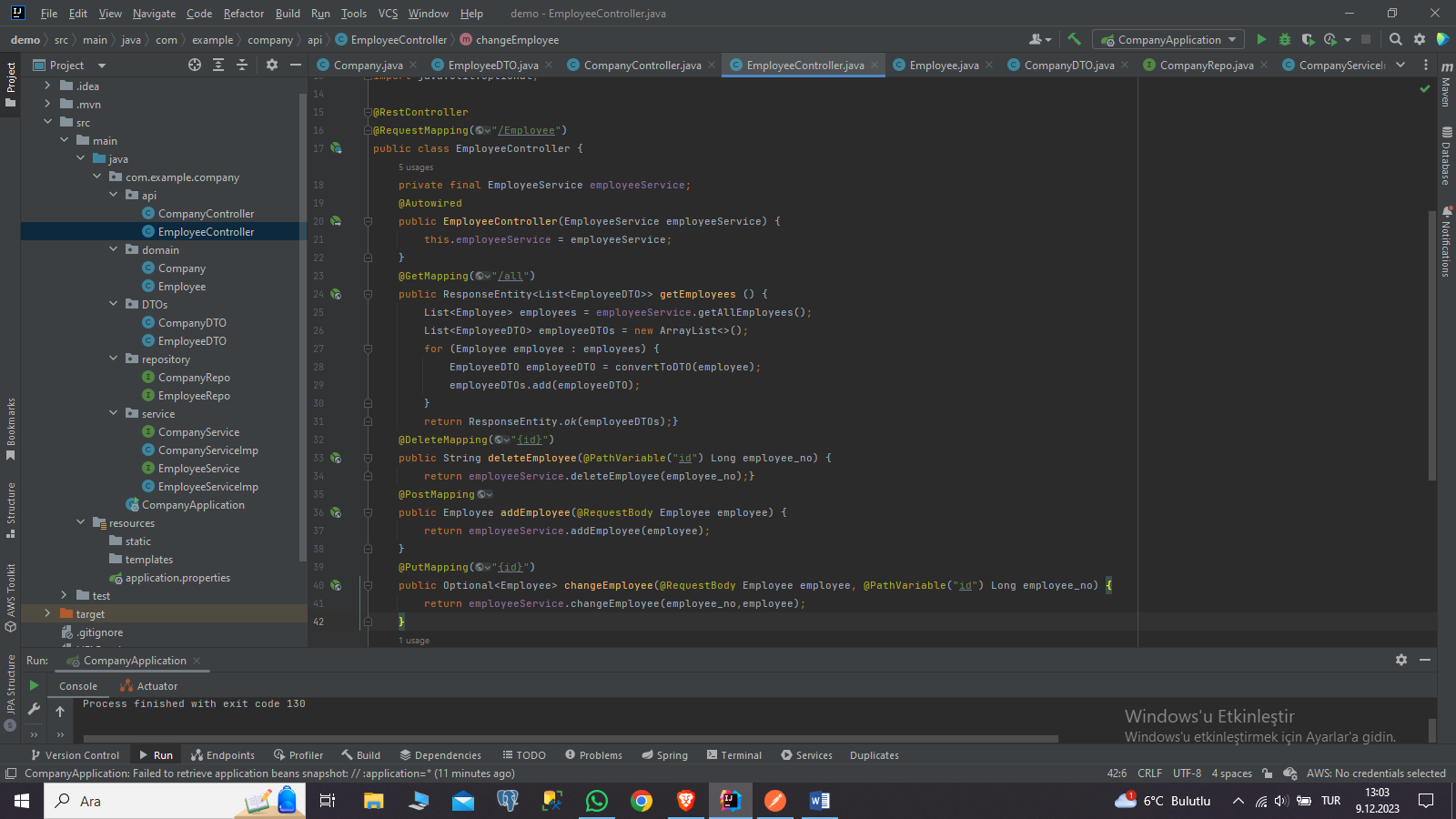


**8-** Employee servisi ve implementasyonu.



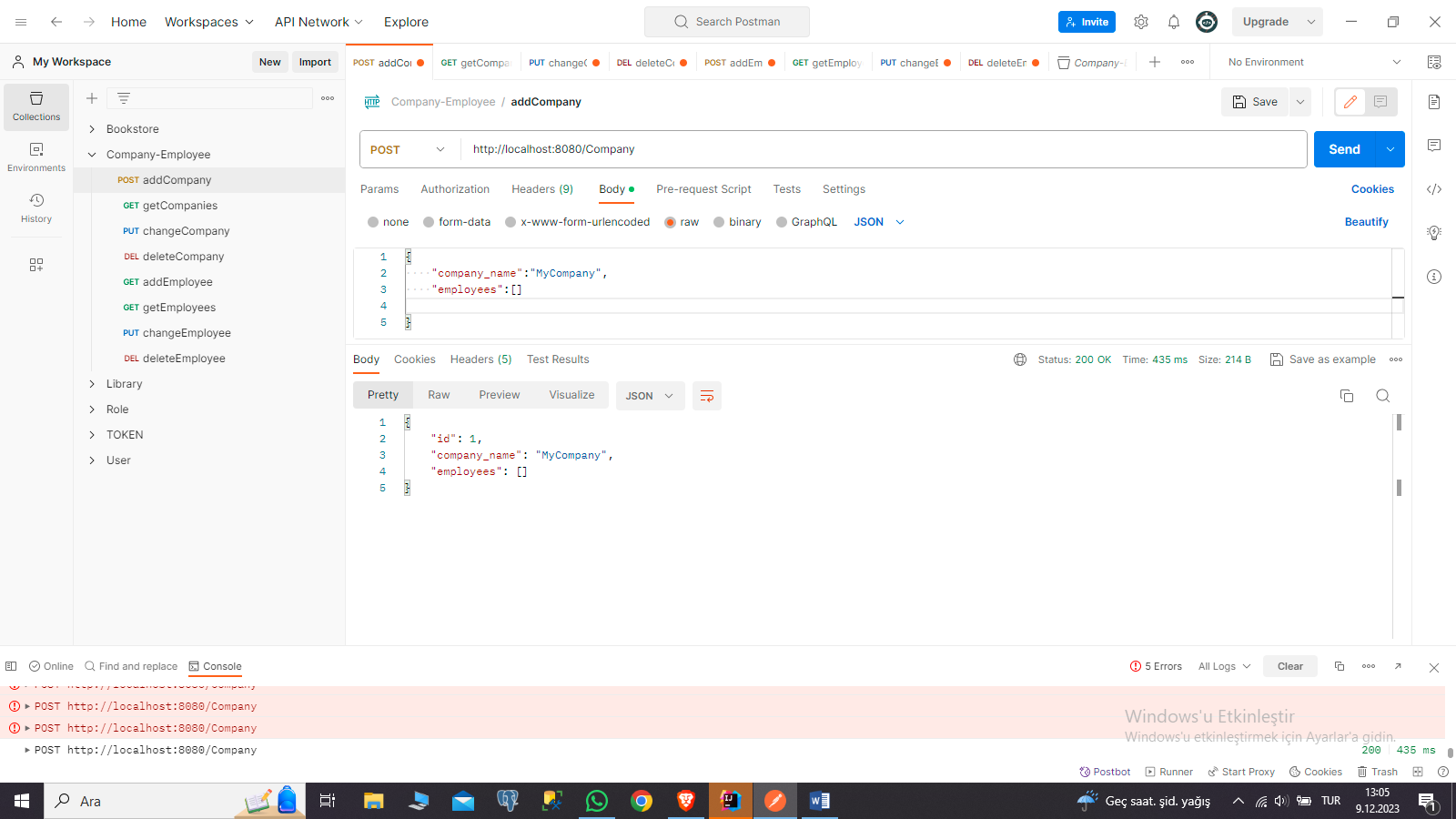
**9-** Son olarakta Controller kısımları. Bu yapılarda, End pointleri uygun biçimde, istenilen http metodlarıyla yapılandırıyoruz. Company için : 

**10-** Employee için :

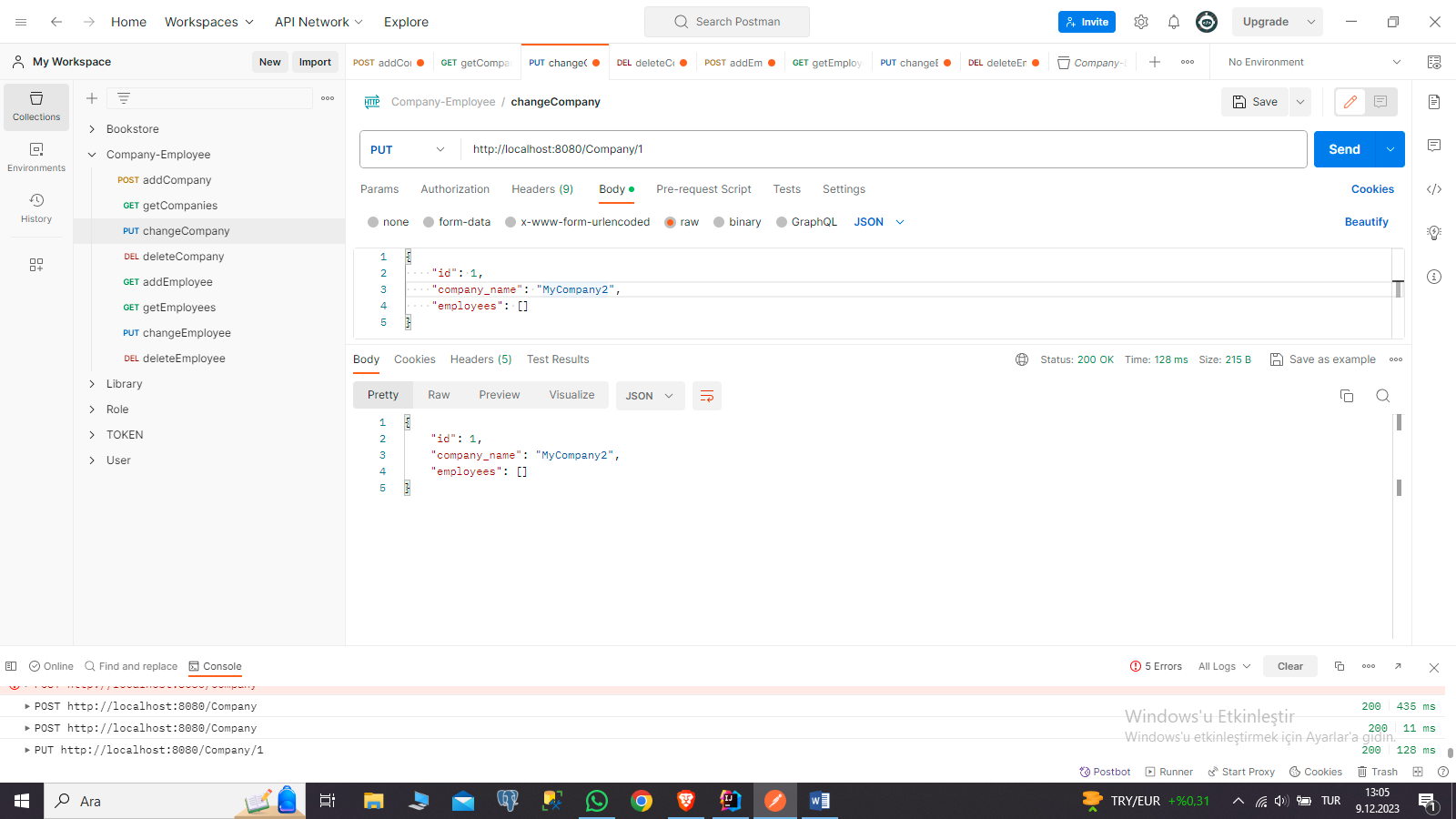


**POSTMAN ÜZERİNDEN TEST**

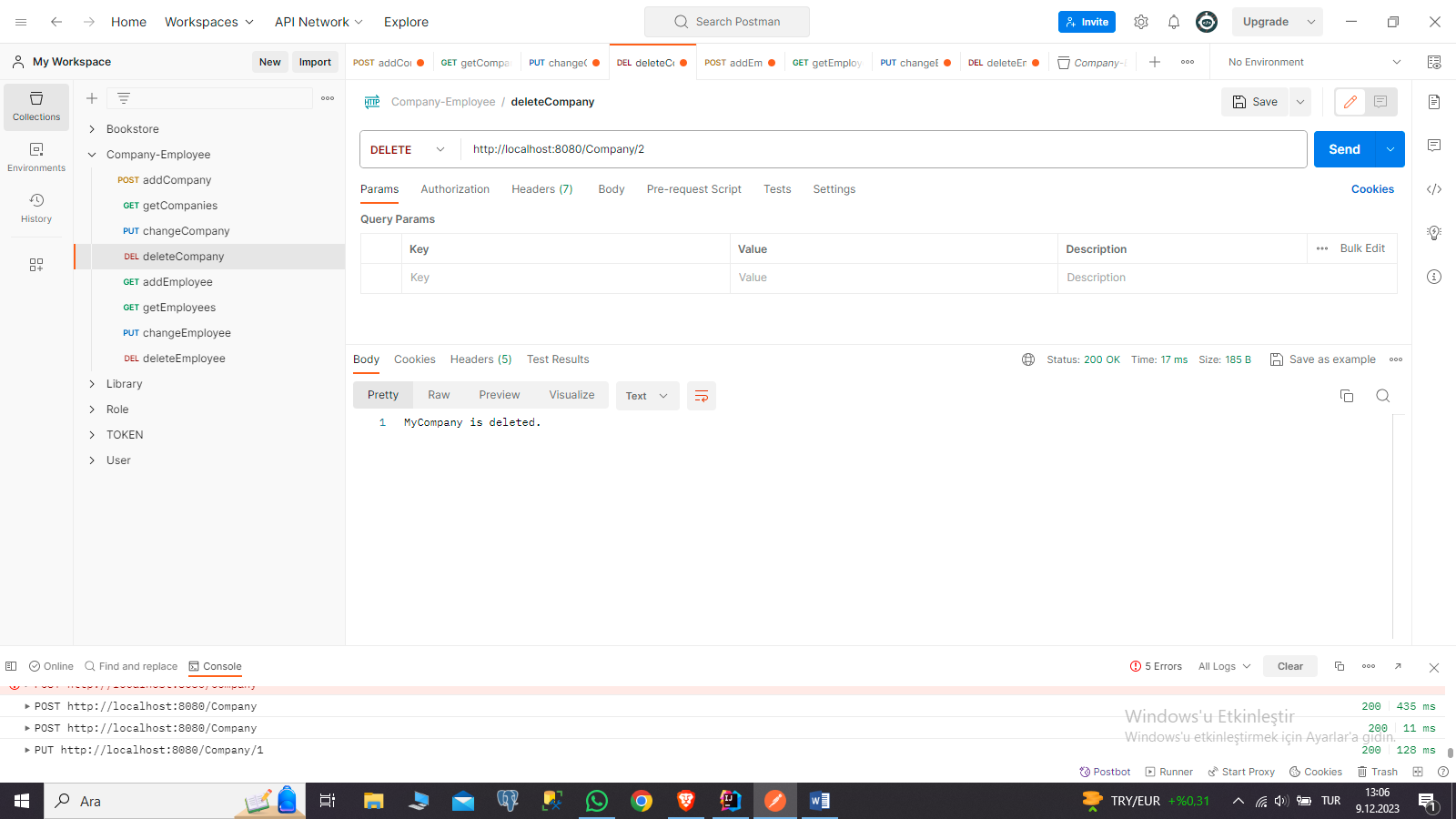
**1-** Company Ekleme



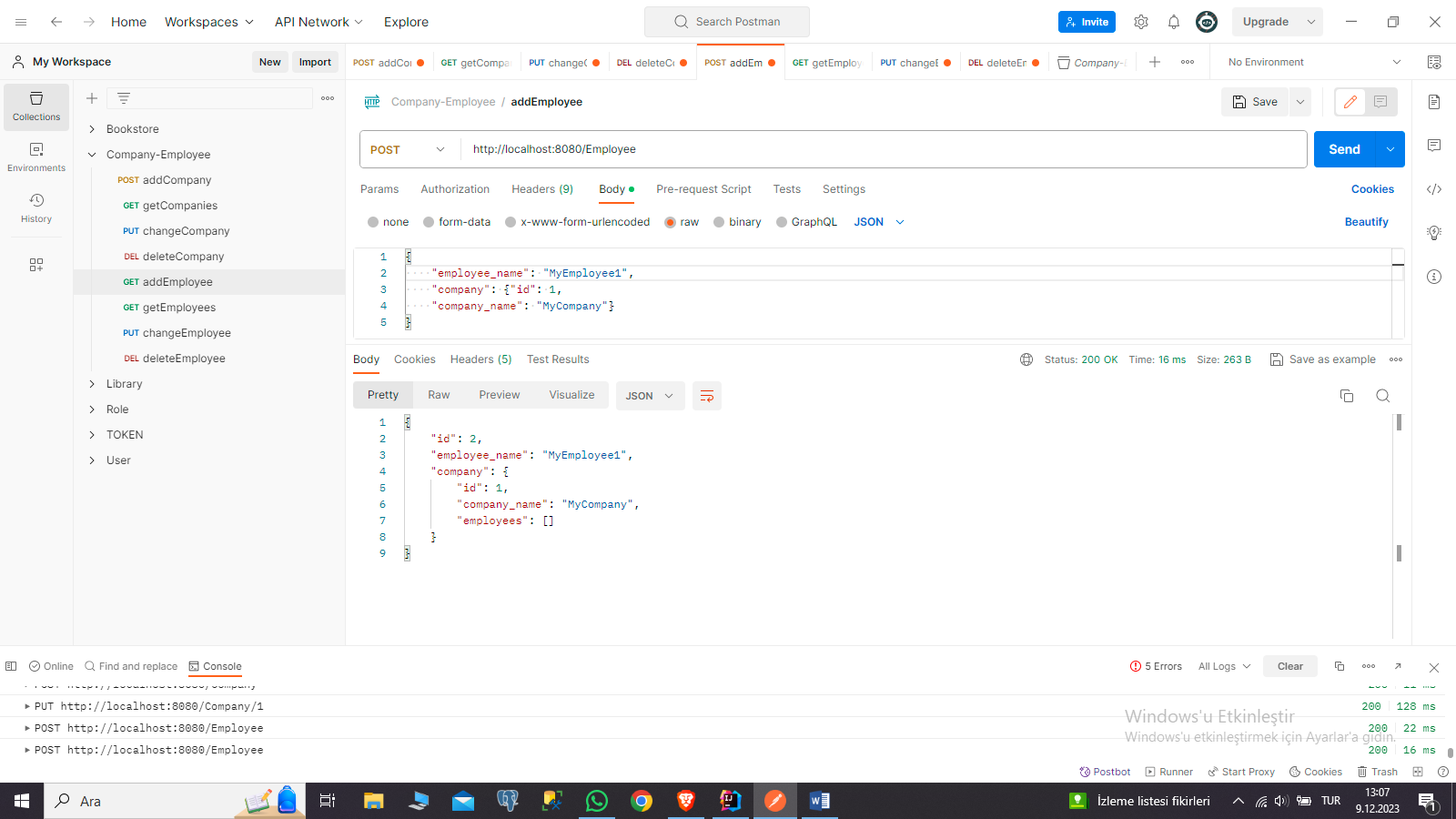
**2-** Company Değiştirme



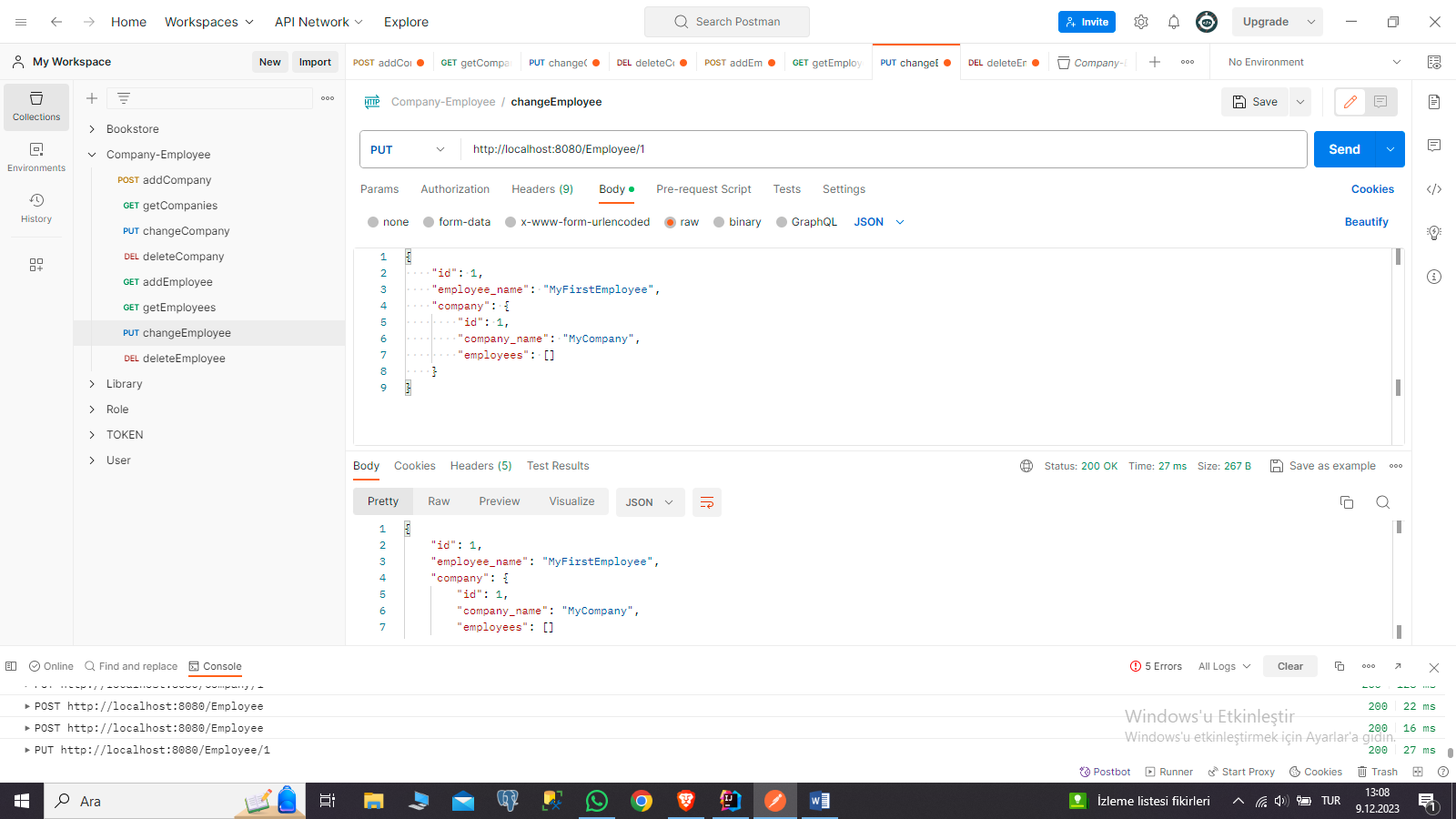
3- Company Silme



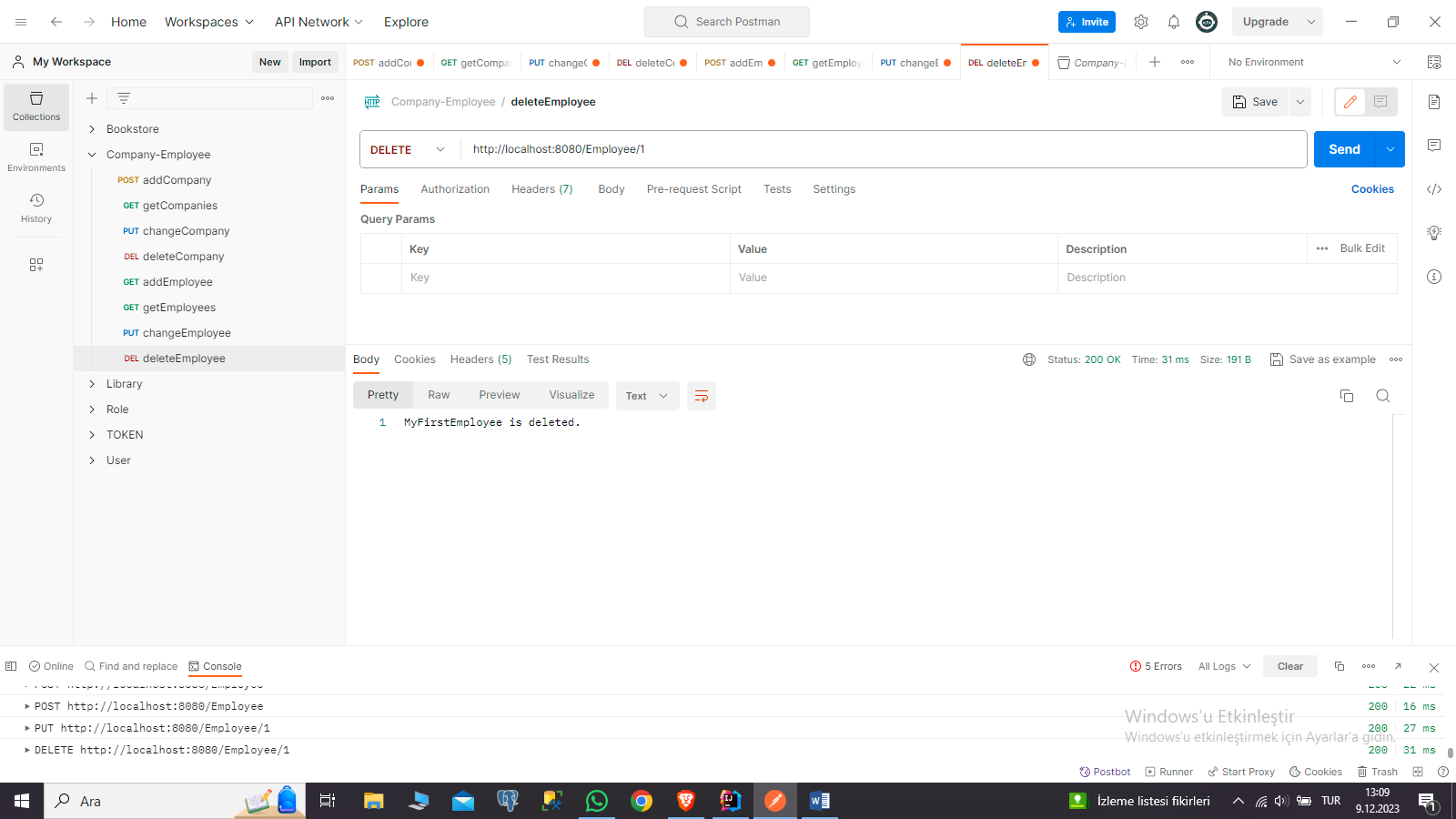
**4-** Employee Ekleme



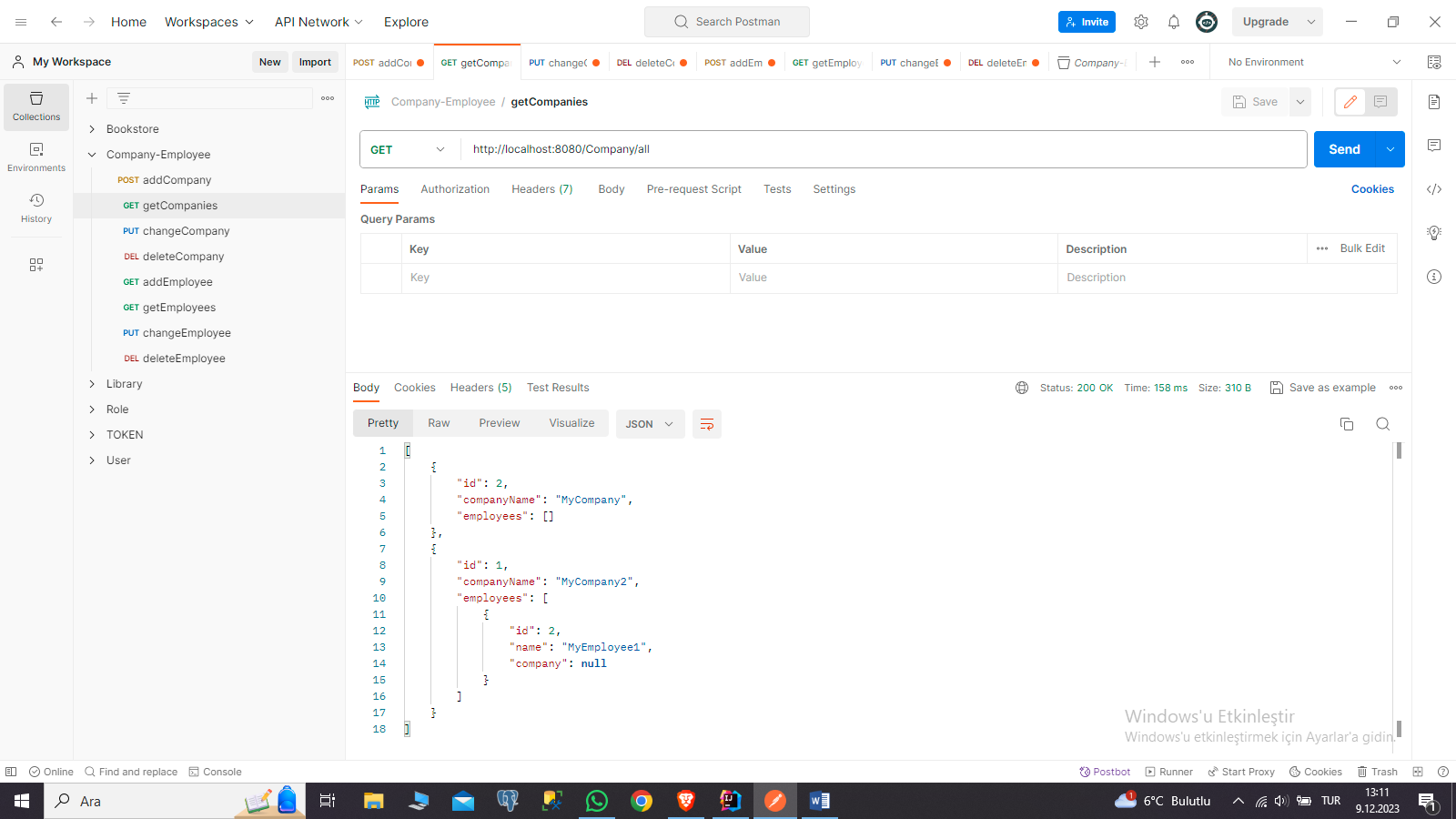
**5-** Employee Değiştirme



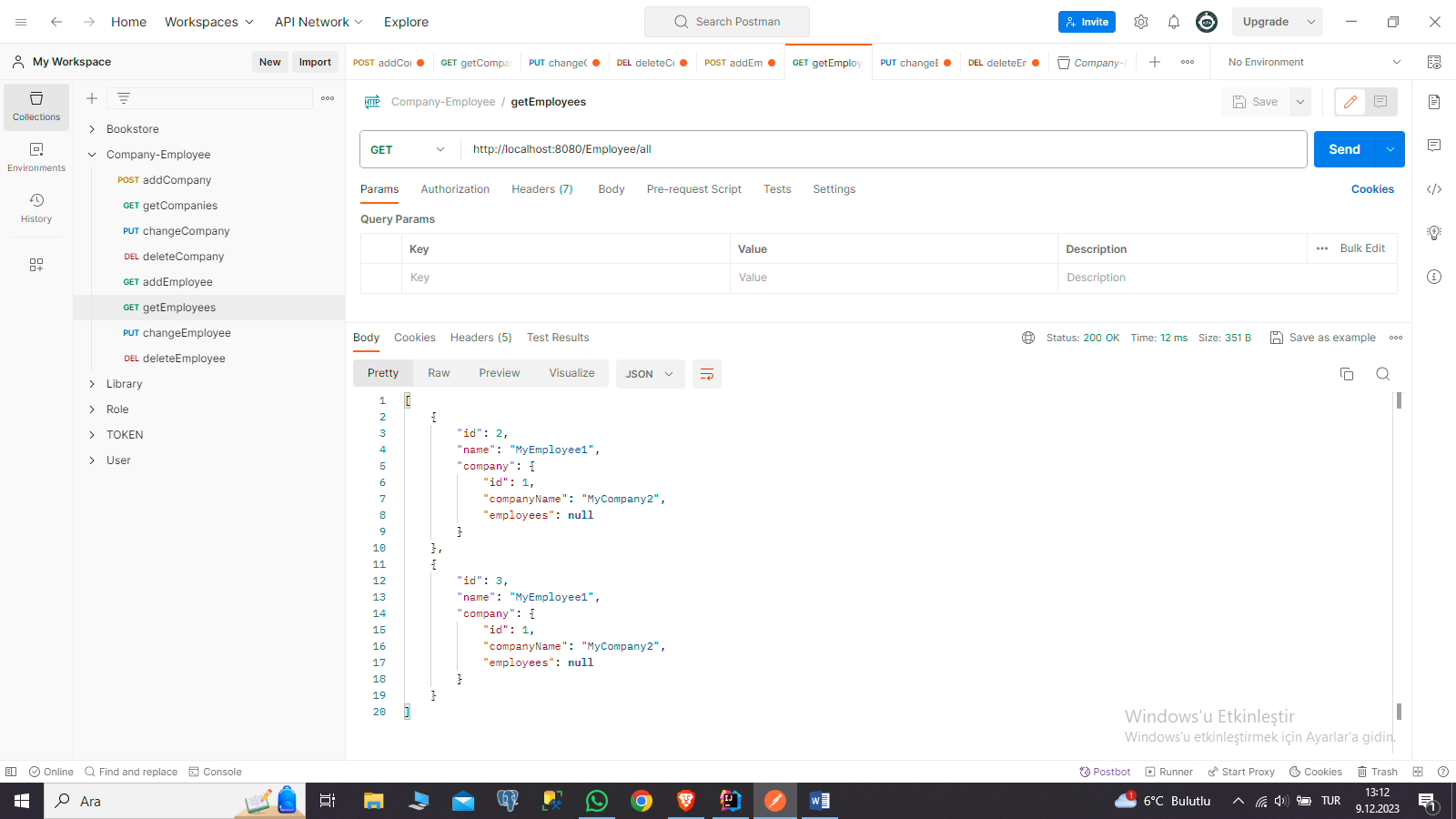
6- Employee Silme



**7-** Tüm Companyleri çekme



**8-** Tüm Employeeleri çekme



Tüm endpointler Postmande test edildi, kaynak kodları Githubta paylaşıldı.