

AdessoRideShare - Proje Detayları

Projede kullanılan Teknolojiler:

- Asp.Net Core 3.1 ile geliştirilmiştir.
- Tüm proje Dockerize edilmiş olup, “docker-compose up” komutuyla ayağa kaldırılabilir.
- Uygulamanın horizontally scaleable olabilmesini kolaylaştırmak için oluşturulan Travel’lar MongoDB veri tabanında tutulmuştur.
- Api testini kolaylaştırmak için Swagger eklenmiştir.
- xUnit kullanılarak domain logic işleten metotlara unit test eklenmiştir.

Uygulamanın Run edilmesi:

Uygulama dockerize edildiği için “docker-compose up” komutuyla ayağa kaldırılmaktadır. Burada önemli nokta Docker’ın çalıştırılan işletim sistemine göre standart driver’lara erişim yetkisi olmasıdır.

Docker:

“docker-compose up” komutuyla .Net Core Api projesi ayağa kaldırılır. MongoDB cloud ortamında çalıştığı için Docker image olarak eklemeyim.

Api’nin Test Edilmesi:

Bu işlem Postman vb bir tool ile veya <http://localhost:49396/swagger/index.html> adresinden SwaggerUI ile yapılabilir.

MongoDB veritabanında oluşturulan Travel objelerini görmek için *appsettings.json* ‘da *connectrionString* olarak da tanımlı olan:

**mongodb+srv://adesso:adesso@cluster0.67qel.mongodb.net/AdessoRideShareDB?retryW
rites=true&w=majority** bağlantısıyla MongoDB Compass üzerinden ulaşabilirsiniz. (Uygulamada Get request Call ederek de Swagger veya Postman üzerinden görülebilirler.)

Notlar

- Entity’leri Controller’a direkt göndermek yerine sadece Controller’da ihtiyaç duyulan alanları kullanmak için DTO’lar oluşturuldu ve DTO’lar ile Entity Objeleri AutoMapper ile maplendi.
- Global Exception Handling için Custom Exception sınıfları ve onları kullanan bir Custom Exception Filter oluşturuldu.
- Api Controller’dan dönmek üzere Custom Response sınıfı oluşturuldu.