

Rent A Car Proje Dokümanı

İÇİNDEKİLER

1. Proje Genel Tanım ve Katmanlar	3
1.1. Genel Tanım	3
1.2. Katmanlar	4
1.2.1. Core Katmanı	4
1.2.2. Data Katmanı	4
1.2.3. Service Katmanı	5
1.2.4. Api Katmanı	5
2. Kimlik Doğrulama Ayarları	6
3. Swagger'dan Api Methodlarına Erişim	7
4. HealthCheck	8

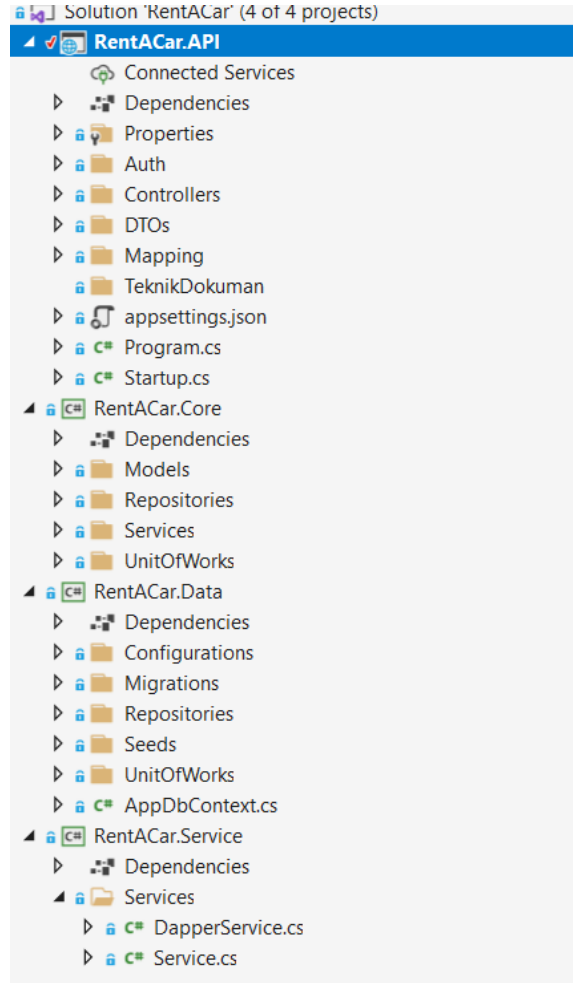
ŞEKİLLER TABLOSU

Şekil 1 Genel Tanım	3
Şekil 2 Core Katmanı	4
Şekil 3 Data Katmanı	4
Şekil 4 Service Katmanı	5
Şekil 5 Api Katmanı	5
Şekil 6 BasicAuth Giriş Ekranı	6
Şekil 7 BasicAuth ile Giriş İşlemi Başarılı	6
Şekil 8 Swagger'da Api Methodları	7
Şekil 9 HealthCheck için Eklenen Kütüphaneler	8
Şekil 10 Veritabanımın HealthCheck Durumu	8

1. Proje Genel Tanım ve Katmanlar

1.1. Genel Tanım

Temel olarak proje; API, Core, Data ve Service olmak üzere 4(dört) katmandan oluşmaktadır. Araba, Alıcı, Kiralama ve Firma olmak üzere 4 (dört) entity üzerine kurgulanmış olup Code First olarak oluşturulmuştur. UnitOfWork ve Repository patternleri kullanılmıştır. Alıcı ve Araba entitylerinin linq, Kiralama ve Firma entitylerinin ise dapper aracılığıyla veritabanı işlemleri gerçekleştirilmiştir. Kimlik doğrulama ayarları Swagger aracılığıyla BasicAuth olarak gerçekleştirilmiştir.

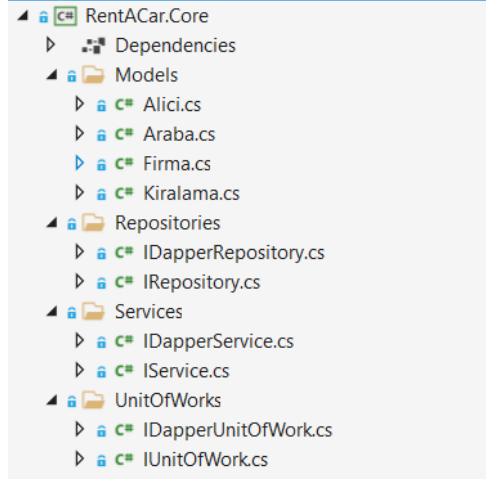


Şekil 1 Genel Tanım

1.2. Katmanlar

1.2.1. Core Katmanı

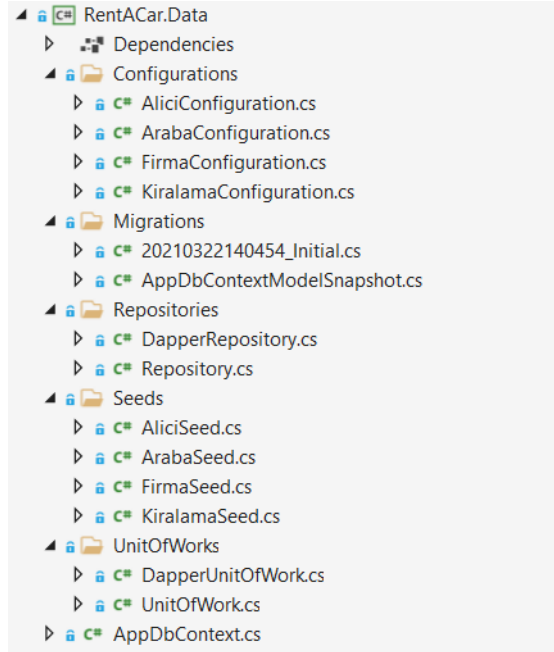
Entity ve interfacelerin bulunduğu katmandır.



Şekil 2 Core Katmanı

1.2.2. Data Katmanı

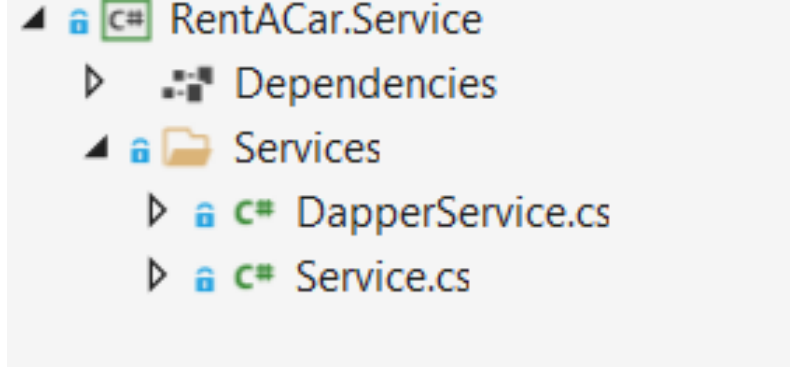
Veritabanıyla haberleşme işlemini gerçekleştiren katmandır. migrationlar, veritabanı oluşturulduktan sonra gelecek default veriler, repositoryler ve db contexti içerir. Core katmanını proje referansı olarak alır.



Şekil 3 Data Katmanı

1.2.3. Service Katmanı

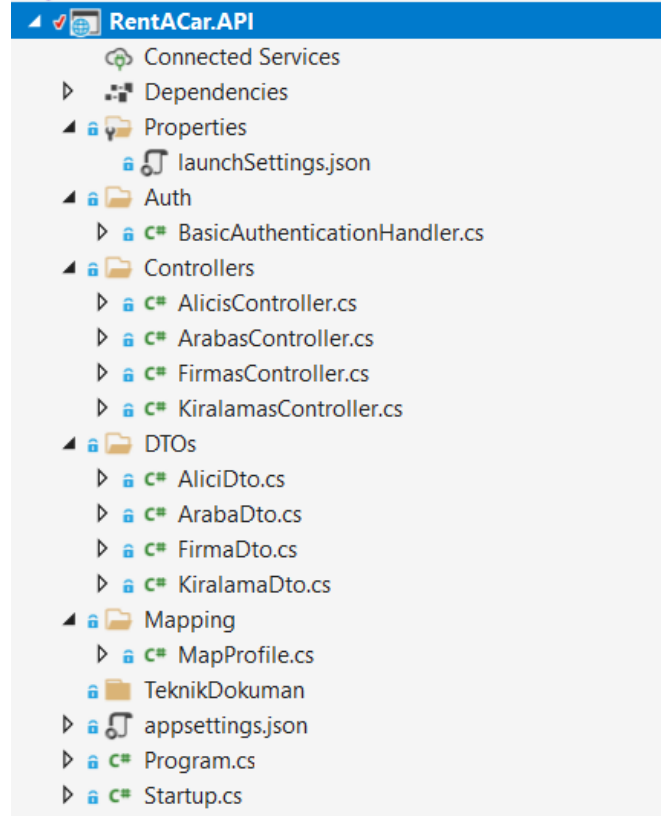
Data katmanından verileri alarak Api katmanında gösterilmesini sağlar. Service sınıflarını, implement dosyalarını, business kodlarını ve dapper'ında bulunduğu katmandır. Core ve Data katmanını proje referansı olarak alır.



Şekil 4 Service Katmanı

1.2.4. Api Katmanı

Projenin temelini oluşturur ve ana işlemlerin gerçekleştiği ve kullanıcının ulaştığı katmandır. Core, Data ve Service katmanını proje referansı olarak alır.



Şekil 5 Api Katmanı

2. Kimlik Doğrulama Ayarları

Kimlik doğrulama ayarlar Swagger aracılığıyla BasicAuth türüyle yapılmıştır. Kullanıcı adı ve şifre bilgisi API katmanında appsettings.json’da, ayarlar ise Startup.cs ve BasicAuthenticationHandler.cs içerisinde tanımlanmıştır.

Available authorizations ×

basic (http, Basic)

Basic Authorization header using the Bearer scheme.

Username:

Password:

Authorize **Close**

Şekil 6 BasicAuth Giriş Ekranı

Available authorizations ×

basic (http, Basic)

Authorized

Basic Authorization header using the Bearer scheme.

Username: Admin

Password: *****

Logout **Close**

Şekil 7 BasicAuth ile Giriş İşlemi Başarılı

Controllerlarda kimlik doğrulama ayarı geçerli olması istenen methodların başına [Authorize] konulması yeterlidir.

3. Swagger'dan Api Methodlarına Eriřim

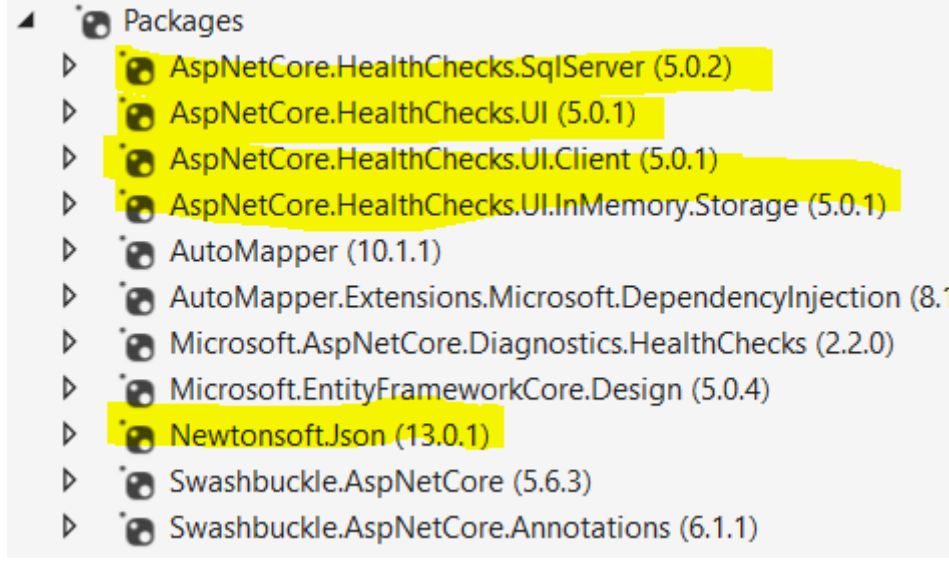
The screenshot displays the Swagger UI for the RentACar.API v1. The interface is organized into sections for different API resources: Alicis, Arabas, Firmas, and Kiralamas. Each section lists five endpoints with their corresponding HTTP methods and descriptions. The endpoints are color-coded: GET (blue), POST (green), PUT (orange), and DELETE (red). A 'Authorize' button is located in the top right corner.

Resource	Method	Endpoint	Description
Alicis	GET	/api/Alicis	Alici bilgilerin listesi verit.
	POST	/api/Alicis	Kullanıcı tarafından girilen bilgilere göre alıcı bilgilerini kaydeder.
	PUT	/api/Alicis	Kullanıcı tarafından girilen bilgilere göre alıcı bilgilerini günceller.
	GET	/api/Alicis/{id}	Kullanıcı tarafından verilen id'ye göre alıcı bilgilerini verit.
	DELETE	/api/Alicis/{id}	Kullanıcı tarafından girilen alıcı id'ye göre seçilen arabayı siler.
Arabas	GET	/api/Arabas	Araba bilgilerin listesi verit.
	POST	/api/Arabas	Kullanıcı tarafından girilen bilgilere göre araba bilgilerini kaydeder.
	PUT	/api/Arabas	Kullanıcı tarafından girilen bilgilere göre araba bilgilerini günceller.
	GET	/api/Arabas/{id}	Kullanıcı tarafından verilen id'ye göre araba bilgilerini verit.
	DELETE	/api/Arabas/{id}	Kullanıcı tarafından girilen araba id'ye göre seçilen arabayı siler.
Firmas	GET	/api/Firmas	Firma ve firmaya bağlı arabaları verit.
	POST	/api/Firmas	Kullanıcı tarafından girilen bilgilere göre firma bilgilerini kaydeder.
	PUT	/api/Firmas	Kullanıcı tarafından girilen bilgilere göre firma bilgilerini günceller.
	GET	/api/Firmas/{id}	Firma ve firmaya bağlı arabaları kullanıcı tarafından verilen id'ye göre verit.
	DELETE	/api/Firmas/{id}	Kullanıcı tarafından girilen firma id'ye göre seçilen firmayı siler.
Kiralamas	GET	/api/Kiralamas	Kiralama ve kiralamaya bağlı arabaları ve alıcıları verit.
	POST	/api/Kiralamas	Kullanıcı tarafından girilen bilgilere göre kiralama bilgilerini kaydeder.
	PUT	/api/Kiralamas	Kullanıcı tarafından girilen bilgilere göre kiralama bilgilerini günceller.
	GET	/api/Kiralamas/{id}	Kiralama ve kiralamaya bağlı arabaları ve alıcıları kullanıcı tarafından verilen id'ye göre verit.
	DELETE	/api/Kiralamas/{id}	Kullanıcı tarafından girilen kiralama id'ye göre seçilen kiralama bilgilerini siler.

Şekil 8 Swagger'da Api Methodları

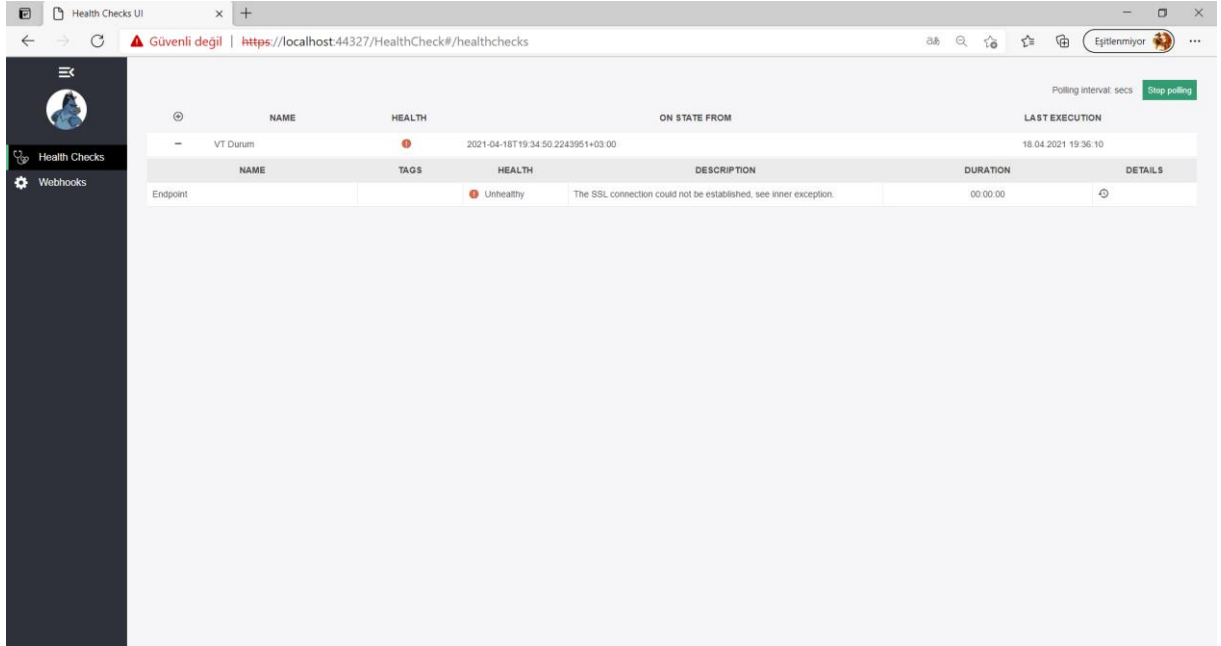
Methodlara Swagger aracılığıyla ulaşılabilirdiği gibi Postman v.b. Api araçlarıyla da ulaşmak, test etmek mümkündür.

4. HealthCheck



Şekil 9 HealthCheck İçin Eklenen Kütüphaneler

Yukarıdaki 4 HealthCheck ve 1 Json paketini API katmanıma ekledim ve ilgili eklemeleri appsettings.json(giriş linki) ve Startup.cs(ayarlar) dosyalarında gerçekleştirdim.



Şekil 10 Veritabanımın HealthCheck Durumu

İlk açılışta SSL'i kabul etmediğim için HealthCheck sonucum unhealthy yani sağlıksız 😊