Car Rental API

In order to easily use this API with the database that you want, I've used an architecture that strongly decouples the Data layer with the Business of the application.
Right now, I've implemented two different data repositories to use the API: Firebase and Internal Data.

Using Firebase

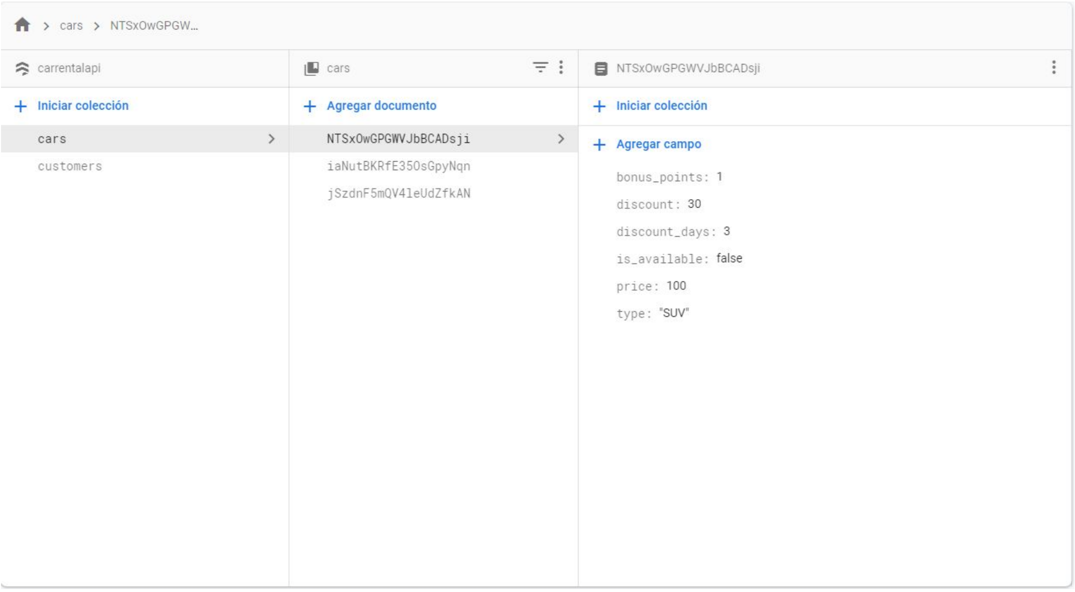
To use firestore with the API you just need to follow these steps:

1. Create a new project in service credentials using the following link: https://console.cloud.google.com/projectselector2/apis/credentials
2. Create and download a new key of your credentials and copy it into the root of the project.
3. In both FirestoreRepository classes in the project, change the PROJECT_ID variable for your actual project_id (e.g. `private const string PROJECT_ID = "carrentalapi";`)

   After create a new project using Firebase console, you should create the following collections and add as much documents as you want:

- cars

  cars {
  type: "SUV",
  price: 100,
  discount: 30,
  discount_days: 3,
  bonus_points: 1,
  is_available: true
  }



NTSxOwGPGWVJbBCADsji
iaNutBKRfE350sGpyNqn
jSzdnF5mQV4leUdZfkAN

bonus_points: 1
discount: 30
discount_days: 3
is_available: false
price: 100
type: "SUV"

- customers

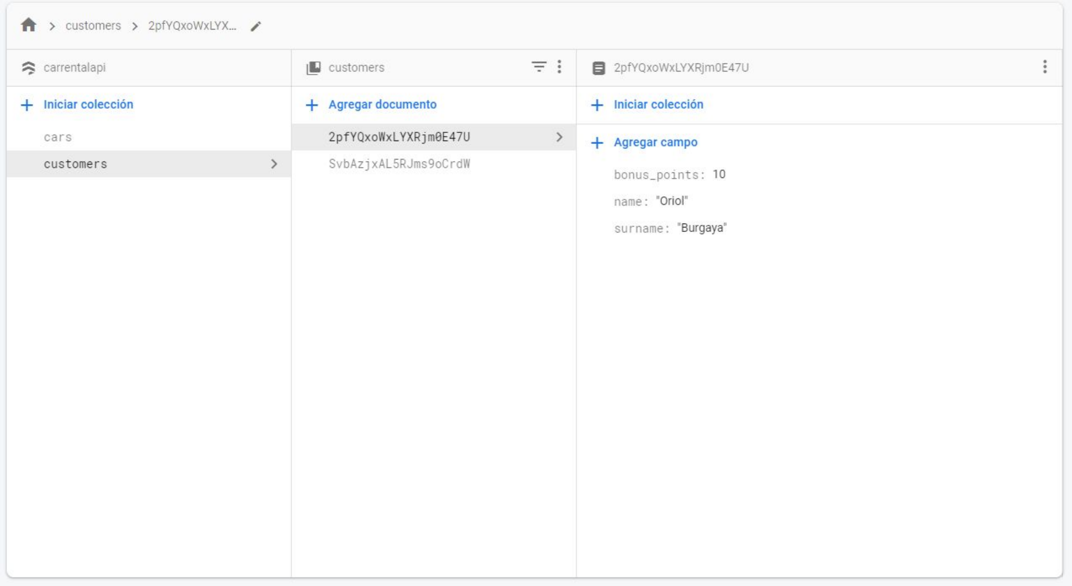
  customers {

  name: “Oriol”,

  surname: “Burgaya”,

  bonus_points: 0

  }



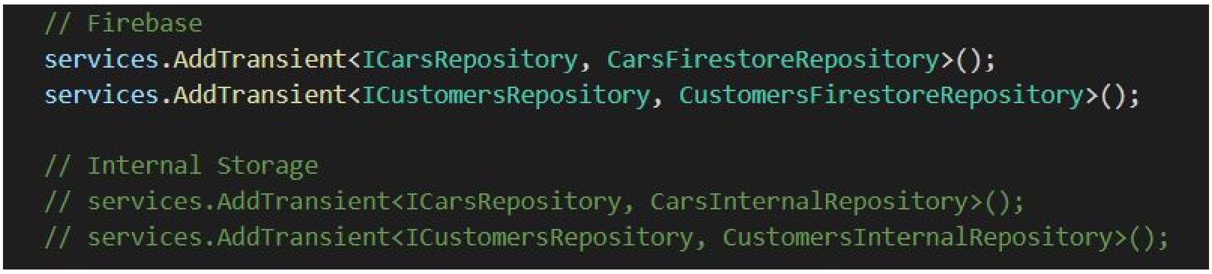
cars
customers

2pfYQxoWxLYXRjm0E47U
SvbAzjxAL5RJms9oCrdW

bonus_points: 10
name: "Oriol"
surname: "Burgaya"

After that, you can use the Firebase repository in the API to use this database.

Be sure that the dependency injection code lines for Firebase repository classes are uncommented.



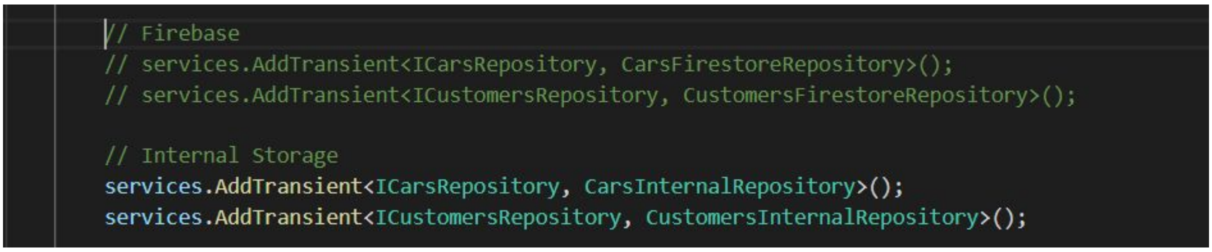
```
// Firebase
services.AddTransient<ICarsRepository, CarsFirestoreRepository>();
services.AddTransient<ICustomersRepository, CustomersFirestoreRepository>();

// Internal Storage
// services.AddTransient<ICarsRepository, CarsInternalRepository>();
// services.AddTransient<ICustomersRepository, CustomersInternalRepository>();
```

Using Internal Data

Because the data is stored internally in data structures, it will be reset every time you execute the program.

To use the program with internal data, you just need to comment the dependency injections of Firebase and uncomment the ones for Internal data. You can found this code lines in the *Startup.cs* file.



```
// Firebase
// services.AddTransient<ICarsRepository, CarsFirestoreRepository>();
// services.AddTransient<ICustomersRepository, CustomersFirestoreRepository>();

// Internal Storage
services.AddTransient<ICarsRepository, CarsInternalRepository>();
services.AddTransient<ICustomersRepository, CustomersInternalRepository>();
```