Testing Notes

Running the project

Requirements

- .NET 8 SDK
- Terminal/Powershell

Steps

On your terminal of choice, navigate to the projects root directory:

```
- 1: Check if .NET is installed:
    - dotnet --version
- 2: Restore all nuget packages:
    - dotnet restore
- 3: Build Application
    - dotnet build
- 4: Run Application
    - Navigate to the sp-back-api directory
    - dotnet run
- 5: Access Application
  - The application should be running and accessing via http://localhost:5000,
testing via swagger is advice for simplification
- Optional: Run tests
  - Open Terminal in the projects root directory
  - dotnet test
```

Testing Data

For simplification purposes, the database has been seeded with 6 vehicles and 3 auctions so that the user has some data to work on if needed. The first auction has already been bided on so that its ready for processing on application start. This auction works as an example for the background processing service.

Vehicles:

```
- Hatchback:
  - Id: 1
  - Model: Hatchback
  - Manufacturer: Mazda
  - StartingPrice: 35000
  - VIN: "VINNUMBER1"
  - NumberOfDoors: 4

- Sedan:
  - Id: 2
```

```
  - Model: Camry
  - Manufacturer: Toyota
  - StartingPrice: 25000
  - VIN: "VINNUMBER2"
  - NumberOfDoors: 10

- Suv:
  - Id: 3
  - Model: Explorer
  - Manufacturer: Ford
  - StartingPrice: 35000
  - VIN: "VINNUMBER3"
  - NumberOfSeats: 10

- Truck:
  - Id: 4
  - Model: Silverado
  - Manufacturer: Chevrolet
  - StartingPrice: 45000
  - VIN: "VINNUMBER4"
  - LoadCapacity: 100

- Sedan:
  - Id: 5
  - Model: Civic
  - Manufacturer: Honda
  - StartingPrice: 28000
  - VIN: "VINNUMBER5"
  - NumberOfDoors: 4

- Suv:
  - Id: 6
  - Model: Model Y
  - Manufacturer: Tesla
  - StartingPrice: 55000
  - VIN: "VINNUMBER6"
  - NumberOfSeats: 4
  -
```

All Vehicles were configured with a Random Production Date in the past based on the systems current date.

Auctions:

```
- Soon to be Completed Auction:
  - Id: 1
  - State: Active
  - Included Vehicles:
    - VehicleId = 1 (Mazda);
    - VehicleId = 5 (Honda)
  - StartTime: CurrentDate - 2 Days
  - EndTime: CurrentDate - 1 Minute
```

```
    - Bids:
      - VehicleId = 1 , Amount = 40000.0 , BidderId = winner1
      - VehicleId = 5 , Amount = 30000.0 , BidderId = winner2

  - Waiting Auction:
    - Id: 2
    - State: Waiting
    - Included Vehicles:
    - StartTime:
    - EndTime:
    - Bids:

  - Active Auction:
    - Id: 3
    - State: Active
    - Included Vehicles:
    - StartTime:
    - EndTime:
    - Bids:
```

Additional Notes

Every functionality in this application is pretty self explanatory and easy to use however keep in mind some notes while testing this application:

- Vehicle Type updates can only be performed if the right attributes are sent for the wanted vehicle type; Sedans cannot contain LoadCapacity values other than 0 in the payload so just exclude any params you don't want to update or use to avoid errors;
- Auction do not have a default end date, and will need to be manually closed via the api;
- Bids can only be done on Active Auctions, so make sure you start the auction before bidding;
- Deleted vehicles are excluded from any Get operation in the vehicle repository with the exception of the search with filters where the user can choose to show deleted vehicles;