# Project structure

There are four projects in the solution. I will explain the purpose of these projects one by one.

## Common

This project will contain data transfer objects and interfaces. This project will contain no business and we will refer this project to all other projects. This is how we can remove the dependency of EFModel on core business. EFModel is a tool which we can change whenever we have better tool available. We can also write mock repositories for unit testing without changing anything in the core business.

## Repository

This project is responsible to communicate with database. We are using EFModel to communicate with the SQL Server for now.

## Engine

This projects will contain the core business logic of the system. We will pass repository interface as a dependency. This project knows nothing how we are keeping our data. For unit tests we can pass mock repositories to this project.

## Api

This is executable project which receives the requests, deals with authentication and authorization related stuff, validate arguments and passes the received information to the Engine. This project is unity container to resolve dependencies.

# Deployment

Create an empty database in SQL Server, I am using SQL Server Express version 14.0.1000.169, and execute DBScript.sql on the newly created database. Enter the details of database in the web.config file of Api project. The name of the connection string in web.config is VehicleTrackingDBEntities.

Open VehicleTracking.sln and compile, on the attempt it will resolve nuget packages. I have removed packages and all other dlls to reduce the size of the zip. Run the solution and click on the Api link on Home page to see auto generated help for the api.

I have added detailed comments in the code. I hope these comments will help you to understand my thought process.

Username and password for the admin user is **admin.** You will have to generate the JWT token to access the functionality for administrator.

# Extensibility

I have added two tables in the database Attribute and AttributeValues. If we want to keep more information about the state of vehicle at the time of recording its position, we can keep it in the above mentioned tables. If those attributes are known at the time development we can more columns in the existing table.

# Scalability

We can spin multiple instances of this service with configured load balancer to divide the load. If single instance of database is not enough to handle the load, database cluster can be used.

# Feedback

Irrespective of hiring and evaluation, I will really appreciate if you can suggest improvements in this architecture.