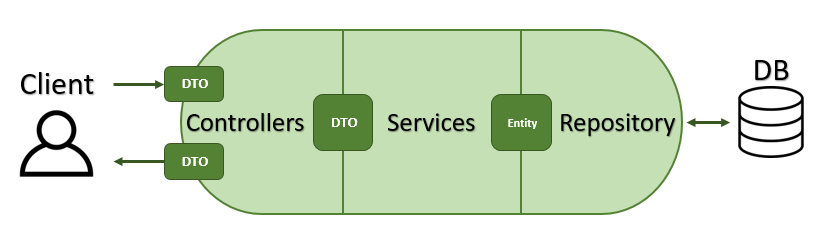
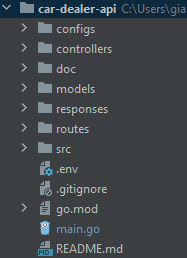
***Golang Test***

***Car Dealer API Rest***

**Initial Criteria**:

1. **IDE to use**: GoLand by NetBrains.
2. **Lenguage**: Golang
3. **Data Base**: the DB to use is "Not Relational", specifically with MongoDB. The name of the database to use is: “carDealerDb”. As a naming convention, “camelCase” is used according to best practices.
4. **Data Base Management**: MongoDB Atlas.
5. **REST API Structure**:



1. **Golang Framework for REST API**: Fiber
2. **Dependencies**:
3. ***github.com/gofiber/fiber/v2***: Framework for building web application.
4. ***go.mongodb.org/mongo-driver/mongo***: Driver for connecting to MongoDB.
5. ***github.com/joho/godotenv***: Library for managing environment variable.
6. ***github.com/go-playground/validator/v10***: Library for validating structs and fields.
7. **Project Folder Structure:**

Where:

**Configs:** this folder is for modularizing project config files

**Controllers:** this folder is for modularizing application logics

**Doc:** this folder is for documentation

**Models:** this folder is for modularizing data and database logics

**Responses:** this folder is for modularizing the response API files

**Routes:** this folder is for modularizing the URL patterns and handler information

**Src:** this folder contains some Fiber and MongoDB Driver functions

**References**

1. Non-Relational Databases Naming Convention: <https://www.coding-guidelines.lftechnology.com/docs/nosql/documentdb/document-db-naming-convention/>).
2. Project Naming Convention: <https://blog.devgenius.io/golang-naming-conventions-72bbaf84e959>
3. Naming Convention de Packages: <https://go.dev/blog/package-names>
4. Best practices to write commits in Git: <https://www.freecodecamp.org/news/writing-good-commit-messages-a-practical-guide/>
5. Git Branching Naming Convention Best Practices: <https://codingsight.com/git-branching-naming-convention-best-practices/>