*This project includes the response to the interview quiz and done by Yarco (*[*https://www.bbish.net*](https://www.bbish.net)*).*

*This document can only be used to verify the skills from this person.*

*Commercial usage or redistribution is forbidden. All rights reserved @2023*

*Resume:*

*En:* [*http://www.bbish.net/files/cv/main.pdf*](http://www.bbish.net/files/cv/main.pdf)

*Cn:* [*http://www.bbish.net/files/cv/main-zh.pdf*](http://www.bbish.net/files/cv/main-zh.pdf)

# Project Intends

Try to use NodeJs to create application which can:

* Allow guests to make/update/cancel reservations
* Allow employees to browser reservations, mark the reservation as completed or canceled

You should use both REST API and GraphQL to finish this.

# Project Building Steps Explanations: Stage-01

## Basic Setup

As we are going to create a nodejs project, the best web framework must be **nestjs** for now. It is better designed and full modular. You could choose “pnpm” as the package management tools.

Graphical user interface, text, application

Description automatically generated

Now, we are going to put this project under version control management system. (Git and Git flow are the basic skills to a developer nowadays). Sometimes, I also use `gi` to automatically add me a .gitignore file. ( I don’t have to do this in this project, for nestjs has already added me one. Just an example using `gi`)

Text, letter

Description automatically generated

Text

Description automatically generated with low confidence

## Task Management

Normally the requirements and tasks are assigned from some task management tools. In this example, I’m going to use local YouTrack to handle this.

Graphical user interface, text, application, Teams

Description automatically generated

Task management explanations:

* “Feature” can be used in BDD
* Those “bold” and/or “italic” texts are emphasized (normally they are “status” or “actions” you should notice)
* Here “class: Reservation” ( and features are actually will be drawn in UML if the business logic is very complicated)

Diagram

Description automatically generated

## IDE

Jetbrains tools are all my favorates. Webstorm is for nodejs project. There are several things I’m going to do first,

### Git Flow

Graphical user interface, text

Description automatically generated

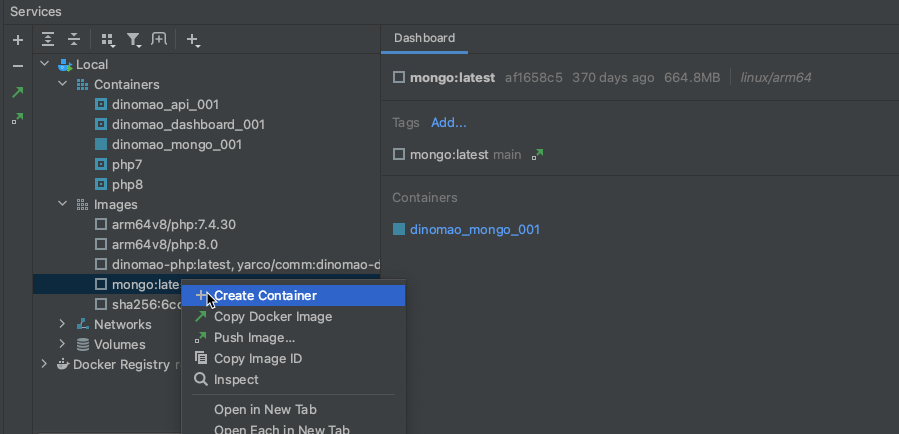
### Task Management Integration

Graphical user interface

Description automatically generated

## Docker

I actually use mongodb (docker version) in this case. But normally, docker will be used as an application container.



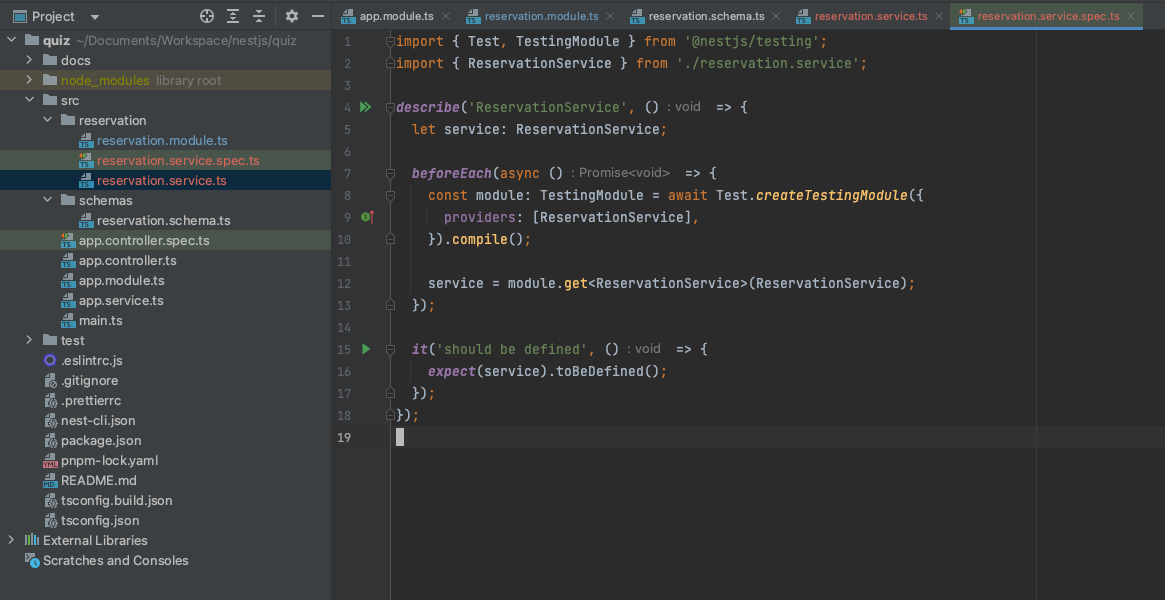
## Database

According to the previous research, we know the main table “Reservation” structure as following (as we are using NoSQL/mongodb, we need to define the schema for it)

Text

Description automatically generated

Then we can generate service file to provide functions on dealing with the database.



According to the functions we’ve already known, we list the actions here:

* Create reservation, extra step: create dto
* Read / List reservation
* Update reservation
* Change status of the reservation

We are going to implement them. And you should also notice, we’ve already create “reservation.service.spec.ts” for unit test (in BDD style)

## Features - Tasks: begin coding

In real world programming, I would use the task management tool to keep the process on track. Graphical user interface, text

Description automatically generated

It works with git flow and youtrack (or any other task management tools)

### task 1: make reservation

We can start from filling \*.spec.ts which is following BDD coding style. And then the function itself.

It sounds like a bug in the test suit. Even use `useValue` to mockup the model data, it still requires real mongodb connection.

Just leave it there, for it doesn’t harm too much.

Text

Description automatically generated

After services and unit tests, we are going to create controller (for REST APIs). Normally, we will add swagger API doc.

The controller and API doc:

Text

Description automatically generated

Just list one as the example:

Graphical user interface, text, application

Description automatically generated

When I’m considering GraphQL, the benefit of using it is commonly known as “flexible client reading”. It does not have so much big improvement on modification and things on admin side. So, here I just do it for the clients – guests.

Graphical user interface, text

Description automatically generated

TODOs:

There are still several parts (backend) I haven’t done yet. But for I want to keep this simple project done in 3 days, I just listed here.

1. Input Data verification
2. Use DotEnv to distinguish different stages
3. Authentication
4. Permission Grants (RBAC vs ABAC)
5. Events (when trigged based on user action)

Below parts will be the frontend stuffs.

## Front End

I’m going to choose “vite” to create the React application.

Graphical user interface, text, application, email

Description automatically generated

And also with Tailwind css framework.Graphical user interface, text, application, email

Description automatically generated

Use similar git and git flow commands to take them into the version control system.

And also, we should turn on CORS as we separated backend and frontend.

Graphical user interface, text, application, email

Description automatically generated

### Simple UI on adding / cancel reservation

* Data listing from GraphQL query
* Cancel op use Axios/post and REST API

Graphical user interface, application, Word

Description automatically generated

Graphical user interface, application

Description automatically generated

I can add more details or make the UI better. Or for example, display human readable text “status:2” equal to “canceled”. But I think it is enough to show my skills in this documentation.

Time consumed: 2.5 Days