Learning:

What is tsconfig.json file and its uses:

Tsconfig.json is special kind of file in TypeScript. Generally, we put this file into Root directory of the project. Tsconfig files specifies root files and compiler options required to compile the project.

A project can be compiled below ways:

1: invoking tsc with no input files, in which case compiler searches for the tsconfig.json file starting from current directory and continuing up the parent directory chain.

2: By invoking tcs with no input files and a –project (or just -p) command line option that specifies the path of a directory containing a tsconfig.json file or a path to a valid .json file containing configurations.

When input files are specified on the command line then tsconfig.json will be ignored.

compilerOptions

it can be omitted then in this case the compilers defaults are used.

If types is specified, only packages listed will be included.

If typeRoots is specified, only packages under typeRoots will be included.

A tsconfig.json file can inherit configurations from another file using the extends property.

Using the "files" property or Using the "include" and "exclude" properties

The "files" property takes a list of relative or absolute file paths. The "include" and "exclude" properties take a list of glob-like file patterns. The supported glob wildcards are:

1: \* matches zero or more characters (excluding directory separators)

2: ? matches any one character (excluding directory separators)

3: \*\*/ recursively matches any subdirectory

So whenever we run the tsc command, all the ts files in the lib folder will be compiled to js files in the dist folder

🡪 Initial Setup:

TypeScript is a typed superset of JavaScript that compiles to plain JavaScript. TypeScript won't do anything in runtime, it works only during compilation time. You will run pure JavaScript files.

Type annotations in TypeScript are lightweight ways to record the intended contract of the function or variable.

TypeScript can offer static analysis based on both the structure of your code, and the type annotations you provide.

* Lets create first route:

Import request and response package from express and create a GET request.

* Now we need to import it into our app.ts file.
* GET: for retrieving data
* POST: for creating new data
* PUT: for updating data
* DELETE: for deleting data