# Web servers. Home task №1

1. **Create** a private GitHub repository.

   *(NOTE:*

   *This repo will be used for other home tasks within this course.*

   *You can use either corporate or personal account.)*

2. **Initiate** an empty npm project.
3. **Create** a simple Node.js web server using either basic http module, or any library or framework that is convenient for you such as express.js, koa.js, nest.js, etc., which will provide the following features:
   a. **Serve** a built Quote single page app (SPA) with all its assets.

   When user opens in browser http://localhost:{APP_PORT}/ URL it must send an **index.html** of built Quote app.

   *(NOTES:*

   *APP_PORT - is a configured port on which app can handle network communication. It must be exposed as environment variable. See notes below.*

   *Source code is attached or can be found here.*

   *App is not built and requires it before your server starts hosting/serving the app.)*

   b. **Provide** a **/ping** health check API endpoint which will return the following json object:

   { “statusCode”: 200, “message”: “OK”, time: “{serverTime}*” }

   * where **{serverTime}** - is a value of server’s time at the moment of response.

   c. **Create** the following REST API endpoints:

   *(NOTE: you can use quotesy npm module or similar with existing list of quotes or create your own one.)*

   *GET* **/api/quotes**

   - returns a list of all available quotes.

   *(NOTE: Where and how to store all data is up to you.)*

   *GET* **/api/quotes/random**

   - returns a random quote from the list.

   *GET* **/api/quotes/random?tag={tag}**

- returns a random quote which has appropriate {tag} from query string or contains appropriate word/phrase int its text.

*POST* **/api/quotes**

- creates a new quote.

*GET* **/api/quotes/:id**

- returns a quote with specified id if exist, otherwise return 404 response.

*PUT* **/api/quotes/:id**

- updates a quote with specified id if exist, otherwise returns 404 response.

*DELETE* **/api/quotes/:id**

- deletes a quote with specified id if exist, otherwise returns 404 response.

(*NOTES*:

*A Postman collection and a sample mock server are attached.*

*Quote model/interface must be as follows:*

```
interface Quote {
  id?: string; // Unique identifier of quote.
  author: string; // Author of a quote.
  text: string; // Quote text.
  source?: string; // optional link/source of quote.
  tags?: string[]; // optional list of tags related to quote.
  createdBy?: string; // app's user who initiate creation of quote.
  createdAt?: string|Date; // timestamp of quote creation.
  updatedAt?: string|Date; // timestamp of quote update.
  isDeleted: boolean; // status of deletion (soft delete).
}
```

*Project structure must be as follows:*

**| - Your-project-folder/**

**| - src/**

**| - app/** - folder with application's code.

**| - main.(js|ts)** - file where application is initialized.

**| - static/** - from this folder server hosts static assets and built Quote app.

**| - client/** - folder with all Quote app's source code.

**| package.json**

**| package-lock.json**

**| README.md**

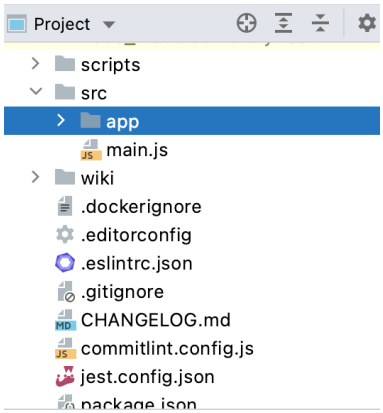
**| .gitignore**

**| .editorconfig**

**| ...**

)

E.g.:




4. **Discover** Quote app project, **install** all necessary dependencies and **try** executing it locally. After that, **add** a necessary environment variable in Quote SPA (see: **src/environments/environment.ts**) to connect it with REST API which you have just implemented. *(NOTE: After update, you will need to re-build your app again. It must make API calls for getting quotes instead of using in-memory ones. It can be checked in Network tab of Developer tools.)*
5. OPTIONAL! **Implement** *graceful shutdown* of your server (look for details HERE and/or HERE).
6. **Store** and **use** the following *environmental variables*:
    a. **APP_PORT** – define an application port on which your server will keep connection and handle requests. E.g., **8080**.
    b. **NODE_ENV** – identifies current application's environment.
    c. **ENV_CONFIGURATION** – identifies a name of current configuration that application server must use (if any).
7. **Add** a necessary npm script(s) to build Quote client and start your server.
8. **Add *README.md*** with instructions of how to setup and run your app.
9. **Add *.gitignore*** file to not track *node_modules/*, *dist/*, *.idea/*, *.vscode/*, *coverage/* folders.
10. **Add *.editorconfig*** file with basic editor settings.
11. **Use** either nodemon or pm2 to start/re-start your app (and track when source code is changed).

*NOTES:*

*It is highly recommended to use GitFlow WorkFlow to deliver changes into your project.*

*We suggest making conventional commits and commit linting* in every app and repository, you work with during this course. You can use Husky npm package to setup appropriate git hooks in your project.

*Please, share an access to your repository with your mentor and other students in your group.*

*When task is done, submit a pull request (PR) and request review from your mentor and other mentees from your group.*

*Duplicate PR's link and attach it in Learn Portal when you submit your work for review.*

*It is recommended to attach screenshots of your work/app along with PR's link (if any).*

*Notify your mentor and other mentees from your group when work is done and ready for review.*

**Evaluation criteria:**

**0** – Nothing has been done.

**3** - Application server is implemented with only **/ping** API endpoint. Static website hosting is not available. Npm script(s) for starting the app is not present. ***README.md*** has not been created. App in genral is working.

**4** - Application server works and serves static application with some issues. Project structure is as it was required. Some configurational files are missing (e.g., no ***.gitignore*** file and all ***node_modules*** are tracked by Git). No npm script(s) is provided or missed ***README.md***. Some (two or more) REST API endpoints have not been implemented.

**5** - All required, and all/some optional requirements are implemented. Application works as expected with no issues and serves static Quote application as well as exposes REST API endpoints.

*NOTE:*

***You can use your personal/own SPA and develop appropriate REST API if needed. Please, contact your mentor or program coordinator beforehand to communicate this option and further steps.***