# Preconditions

To run a project located in the zip file attached to the home task follow the following steps.

Run project:

* “./server” – run “npm i”;
* “./web” – run “npm i”;
* “./server” – run “npm start”;
* open browser on <https://localhost:3000>

# Task description

The goal of this task is to refactor the web part of the application. You will be needed to refactor the code from JS to TS and make it testable. After it you should cover it with unit tests.

\*The home task should be implemented in Typescript.

# Evaluation criteria

For every day of lateness there is a penalty in 0.5 point.

Maximum 10 points.

1. Set-up webpack config to compile TS code [2 points]
   1. It should contain webpack-dev-server
   2. TS code should be minified and transpiled into ES5
   3. Use HtmlWebpackPlugin to include bundled code into index.html file
   4. You can use any CSS pre-process if you wish.
   5. jQuery should be explicitly imported rather than add to the global scope via script in HTML page.
   6. Don’t forget to change the path to index.html file in server part to make sure back-end will server build static files correctly.
2. Code should be refactored to become testable. Currently, it is not possible to test it, because all the code is in single function-listener [4 points]
   1. The logic should be moved to a Class that should expose public methods to set-up listeners and etc. The constructor should accept the jQuery instance.
   2. All listeners should be moved to the Class methods so that you could test them later.
   3. Ajax requests should be moved to a dedicated class-service, which accepts the jQuery object as a constructor parameter.
   4. The Class should accept the class-service via constructor.
   5. Feel free to apply any other approaches that will make code more maintainable, readable, and testable.
   6. **The logic of the application should be the same as before the refactoring!**
3. Cover existing functionality with unit tests [4 points]
   1. Write unit tests for the Class.
   2. The business logic and edge cases should be covered in tests.
   3. Stub jQuery object that you pass to the class instance, to avoid manipulation with DOM tree.
   4. Stub class-service that performs Ajax requests during the testing of the class to avoid real BE interactions.
   5. Write unit tests for class-service. Validate that it performs correct requests. You can stub jQuery object for that purpose.