Coding Standard

MonoToneID

May 2019

1 Introduction

This document will discuss the coding standards practised in our project to help ensure that our code is consistent. It will also discuss the tools used to help ensure the non-violation of our coding standards.

2 Languages

The languages we use in our project are Typescript and Java. Typescript is used by Angular for front-end development.Our back-end uses Spring Boot which is based on Java.

3 Custom Rules

For our project we name variables and functions using descriptive names, however the name must contain a maximum of 6 consecutive words.

We strive to write code that is easy to understand by avoiding using obscure constructs and "too clever code"(code that takes a long time to understand). examples of this would be using a ternary operator instead of a simple if statement.

The rules in this section are not enforced by tools rather by the team members through code reviews.

4 Tools

The environment where we write our code is Visual Studio Code which allows us to install extensions that automatically look for violations as we code.

The first of the extension that we use in our front-end implementation is TSLint. The rules we use are provided by Angular in a tsconfig.ts file. TSLint scans the code and highlights the violations in the code.

Another extension we use is Checkstyle for our back-end implementation. For the back-end coding styles rules we use google's checkstyle configuration file found at (`https://github.com/checkstyle/checkstyle/blob/master/src/`

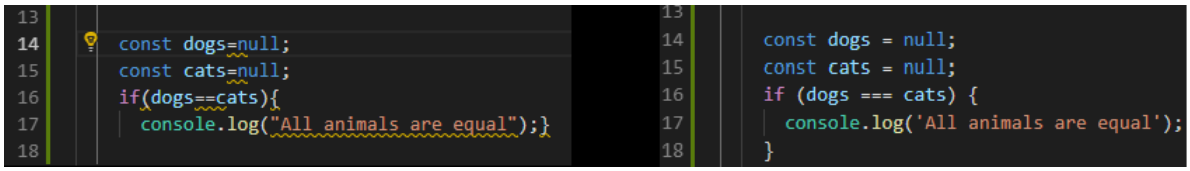

Figure 1: TSLint example.

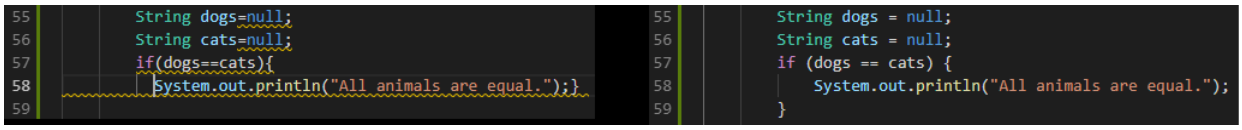

Figure 2: Checkstyle example.

`main/resources/google_checks.xml`) . The extension checkstyle functions similar to TSLint, it scans the code files and highlights the violations and classifies them as warnings. Checkstyle can also be configured to cause a build fail when violations exist in the code.

These tools are not the only way one can check coding standards violations. For Angular one can simply run the command ng lint, and for Spring Boot the command gradlew check or gradlew build.

The Checkstyle package also produces html reports of the violation, found in the build folder, as can be seen in Figure 3.

5 Repository Folder Structure

Our consists of 6 folders:

- Backend
- Frontend
- Database
- Device Simulations
- PV System Simulation
- Documentation

The repository also contains a yaml file for TravisCI and Readme file. Figure 3. Shows the structure of the backend folder. Since we are using a layered architecture, our code files are organized in terms of layers. Figure 4. Shows the structure of the frontend folder. The folders in the frontend are also organized according to our architecture. The Database folder contains the SQL script used to create the database tables. The Documentation folder contains all our documentations and presentations.

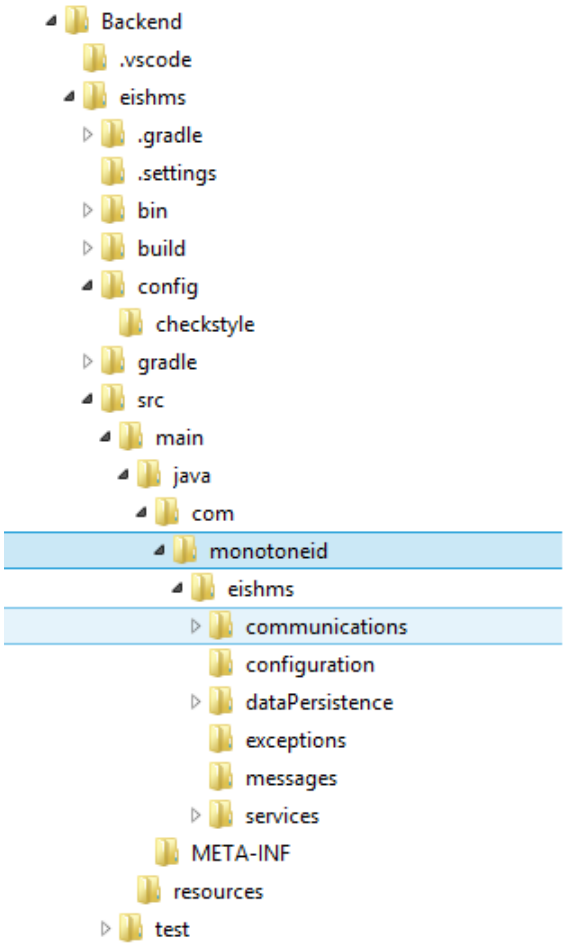

Figure 3: Backend Folder Structure

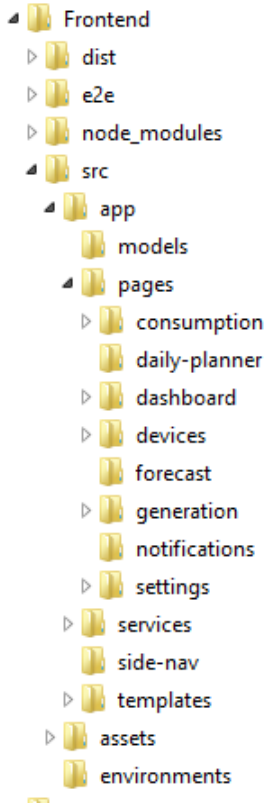

Figure 4: Frontend Folder Structure