# 

# **FHN Platform Repositories & Coding Guidelines**

The following resources should be used for code contributed to the FHN project. Overall, the goal is to provide effective, high quality code that is maintainable, modular, testable, and modern.

## 

**Project Github Repository**

<https://github.com/momentumdesignlab/dev-mdl-fhn>

## **Branching**

Source control should follow [Gitflow practices](https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow).

**Coding Guidelines**

The JavaScript Standard Style

<https://standardjs.com/>

The TypeScript Standard Style  
<https://github.com/standard/ts-standard>

The Angular Style Guide

<https://angular.io/guide/styleguide>

## **LINT Management**

Linting of repository files is mandatory. The lint engine used is [tslint](https://palantir.github.io/tslint/) (by default) and the lint rules will be stored under source control. For tslint the rules are found in tslint.json. It is imperative that lint rules are common across team members. Default linting rules can be tuned to the teams preferences.

## 

## **Code Formatting**

Code formatting is managed by the [prettier subsystem](https://prettier.io/) (by default). The formatting rules are found as src/.prettierrc under source control. The formatting rules should be common among collaborators to keep a consistent appearance. The code format should be kept in agreement with the lint rules.

## **Error Handling**

The best case for error management is a centralized error stream. This provides for low troubleshooting overhead (no hunting for scattered log files) as well as the option of a global repository system (like [Sentry](https://sentry.io/welcome/)) that can be monitored to catch scattered events.

## **Test Considerations**

Screen elements should be made with unique ids so they can be attained by the test framework. This allows for automated testing and proper integration with upstream continuous integration/deployment. Using an id attribute is required. In rare cases where ids are not possible (due to component generated HTML), targeting ids should be wired into parent or child elements to provide anchors for test infrastructure.

**XPath-based testing is brittle and expensive to maintain and can be avoided by utilizing simple unique element ids.**

|  |
| --- |
| <**div** id="divElement1" v-on:click="select($event)">...</**div**> |

For reference, here is a good list of CSS selection strategies.

<https://www.w3schools.com/cssref/css_selectors.asp>