TURING-GALLERY-FRONT

VERSION 1.0

CODE ANALYSIS

By: default

2020-01-27

CONTENT

Content 1

Introduction 2

Configuration 2

Synthesis 2

Metrics 2

Volume 3

Issues count by severity and type 3

Charts 4

Issues 5

INTRODUCTION

This document contains results of the code analysis of TURING-GALLERY-FRONT.

CONFIGURATION

- Quality Profiles
  - Names: Sonar way [CSS]; Sonar way [TypeScript]; Sonar way [HTML];
  - Files: AW-vB-6R6MX5uumeTY8S.json; AW-vB_h16MX5uumeTZp4.json; AW-vB_Ue6MX5uumeTZfB.json;
- Quality Gate
  - Name: Sonar way
  - File: Sonar way.xml

SYNTHESIS

Quality Gate	Reliability	Security	Maintainability	Coverage	Duplication
ERROR	B	A	A	0.0 %	0.0 %

METRICS

	Cyclomatic Complexity	Cognitive Complexity	Lines of code per file	Comment density (%)	Coverage	Duplication (%)
Min	0.0	0.0	0.0	0.0	0.0	0.0
Max	123.0	24.0	2629.0	80.0	0.0	0.0

TURING-GALLERY-FRONT

VOLUME

Language	Number
CSS	1244
TypeScript	712
HTML	673
Total	2629

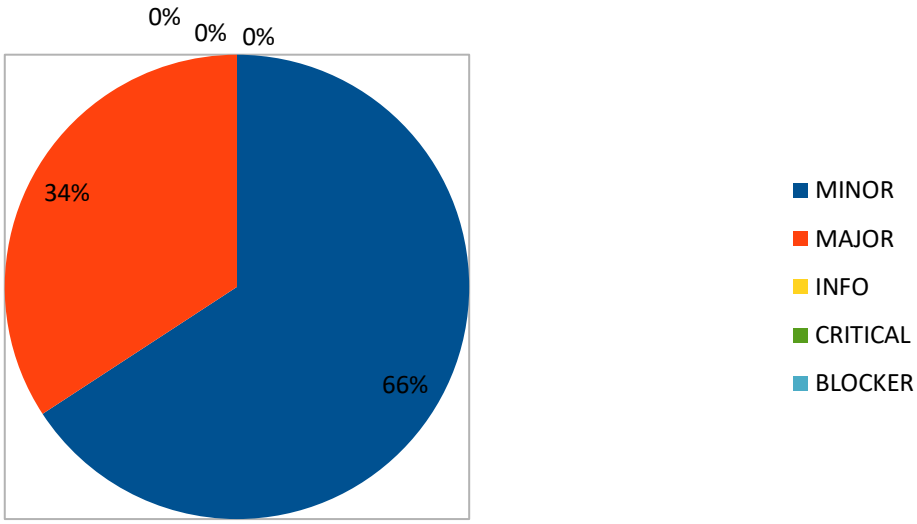
ISSUES COUNT BY SEVERITY AND TYPE

Type	Severity	Number
VULNERABILITY	BLOCKER	0
VULNERABILITY	CRITICAL	0
VULNERABILITY	MAJOR	0
VULNERABILITY	MINOR	0
VULNERABILITY	INFO	0
BUG	BLOCKER	0
BUG	CRITICAL	0
BUG	MAJOR	0
BUG	MINOR	24
BUG	INFO	0
CODE_SMELL	BLOCKER	0
CODE_SMELL	CRITICAL	0
CODE_SMELL	MAJOR	13
CODE_SMELL	MINOR	1

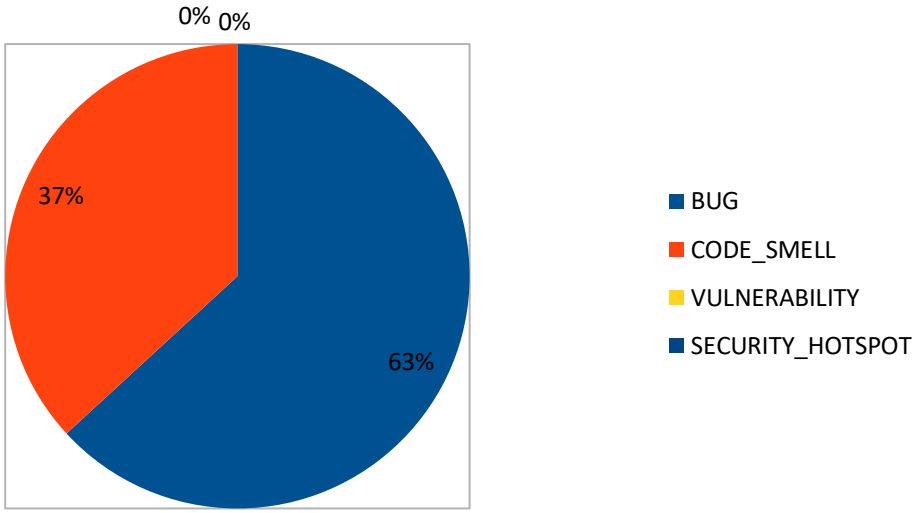
CODE_SMELL	INFO	0
SECURITY_HOTSPOT	BLOCKER	0
SECURITY_HOTSPOT	CRITICAL	0
SECURITY_HOTSPOT	MAJOR	0
SECURITY_HOTSPOT	MINOR	0
SECURITY_HOTSPOT	INFO	0

CHARTS

Number of issues by severity



Number of issues by type



ISSUES

Name	Description	Type	Severity	Number
"" and "" tags should be used	The / and /<i> tags have exactly the same effect in most web browsers, but there is a fundamental difference between them: and have a semantic meaning whereas and <i> only convey styling information like CSS. While can have simply no effect on a some devices with limited display or when a screen reader software is used by a blind person, will: Display the text bold in normal browsers Speak with lower tone when using a screen reader such as Jaws Consequently: in order to	BUG	MINOR	24

	convey semantics, the and <i> tags shall never be used, in order to convey styling information, the and <i> should be avoided and CSS should be used instead. Noncompliant Code Example <i>car</i> <!-- Noncompliant --> train <!-- Noncompliant --> Compliant Solution car train Exceptions This rule is relaxed in case of icon fonts usage. <i class="..." aria-hidden="true" /> <!-- Compliant icon fonts usage -->			
Sections of code should not be commented out	Programmers should not comment out code as it bloats programs and reduces readability. Unused code should be deleted and can be retrieved from source control history if required.	CODE_SMELL	MAJOR	2
Selectors should not be duplicated	Duplication of selectors might indicate a copy-paste mistake. The rule detects the following kinds of duplications: within a list of selectors in a single rule set for duplicated selectors in different rule sets within a single stylesheet. Noncompliant Code Example .foo, .bar, .foo { ... } /* Noncompliant */ .class1 { ... } .class1 { ... } /* Noncompliant */ Compliant Solution .foo, .bar { ... } .class1 { ... } .class2 { ... }	CODE_SMELL	MAJOR	5
CSS files should not be empty	This rule raises an issue when a CSS file is empty (ie: containing only spaces).	CODE_SMELL	MAJOR	6
Extra semicolons should be removed	Extra semicolons are usually introduced by mistake, for example because: It was meant to be replaced by one more property declaration, but this was forgotten. There was a typo which lead the semicolon to be doubled, i.e. ;;. See MISRA C:2004, 14.3 - Before preprocessing, a null statement shall only occur on a line by itself; it may be followed by a comment provided that the first character following the null statement is a white-space character. MISRA C++:2008, 6-2-3 - Before preprocessing, a null statement shall only occur on a line by itself; it may be followed by a comment, provided that the first character following the null statement is a white-space character. CERT, MSC12-C. - Detect and remove code that has no effect or is never executed CERT, MSC51-J. - Do not place a semicolon immediately following an if, for, or while condition CERT, EXP15-C. - Do not place a semicolon on the same line as an if, for, or while statement	CODE_SMELL	MINOR	1