CTS-REACT-PROJECT

VERSION 1.0

Code analysis

By: Administrator

2023-06-07

CONTENT

INTRODUCTION

This document contains results of the code analysis of CTS-react-project.

CONFIGURATION

- Quality Profiles
  - Names: Sonar way [CSS]; Sonar way [JavaScript]; Sonar way [HTML];
  - Files: AYiUIT3lJckAtlJ3-wmf.json; AYiUIUB7JckAtlJ3-w1X.json; AYiUIUY0JckAtlJ3-xoJ.json;
- Quality Gate
  - Name: Sonar way
  - File: Sonar way.xml

SYNTHESIS

ANALYSIS STATUS

Reliability	Security	Security Review	Maintainability
A	A	A	A

QUALITY GATE STATUS

Quality Gate Status	Passed

Metric	Value
Reliability Rating on New Code	OK
Security Rating on New Code	OK
Maintainability Rating on New Code	OK
Coverage on New Code	OK
Duplicated Lines (%) on New Code	OK
Security Hotspots Reviewed on New Code	OK

METRICS

Coverage	Duplication	Comment density	Median number of lines of code per file	Adherence to coding standard
0.0 %	0.0 %	8.7 %	25.0	100.0 %

TESTS

CTS-react-project

Total	Success Rate	Skipped	Errors	Failures
0	**0 %**	**0**	**0**	**0**

DETAILED TECHNICAL DEBT

Reliability	Security	Maintainability	Total
-	-	0d 0h 10min	**0d 0h 10min**

METRICS RANGE

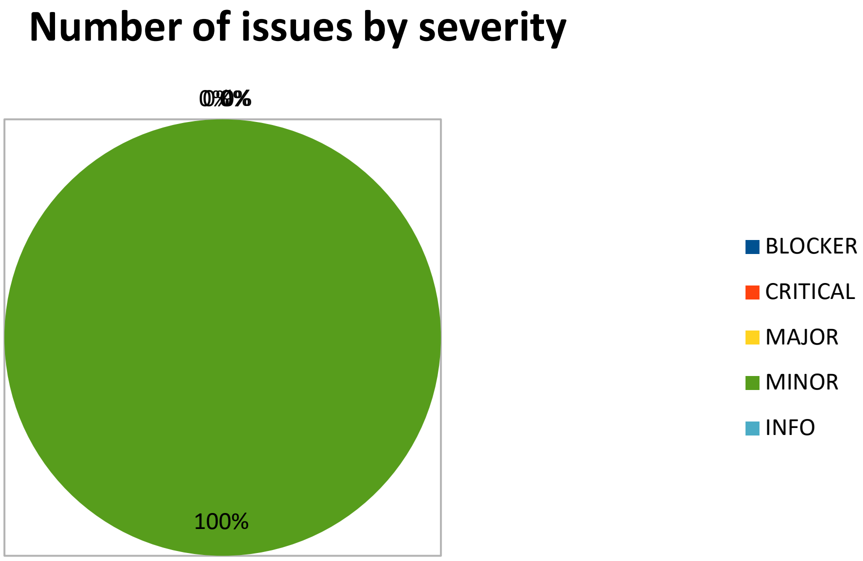
	Cyclomatic Complexity	Cognitive Complexity	Lines of code per file	Comment density (%)	Coverage	Duplication (%)
Min	0.0	0.0	4.0	0.0	0.0	0.0
Max	26.0	16.0	304.0	50.0	0.0	0.0

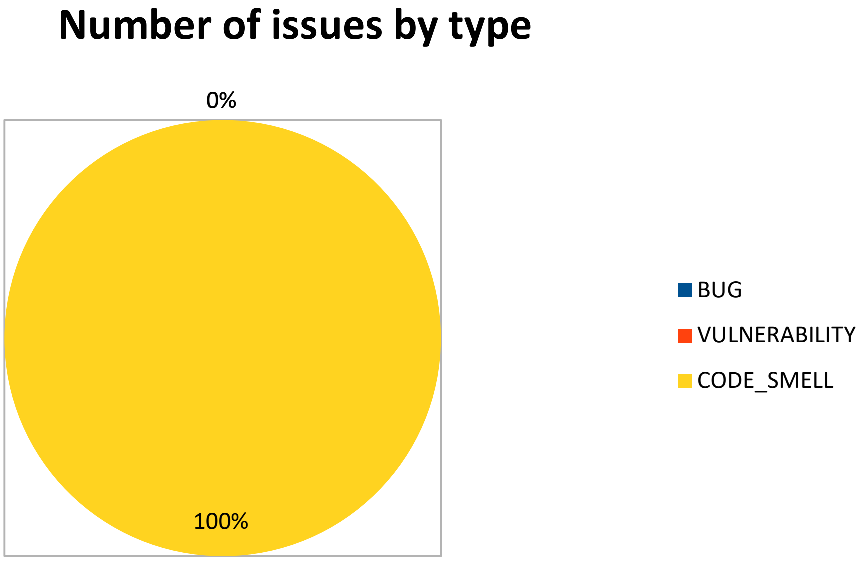
VOLUME

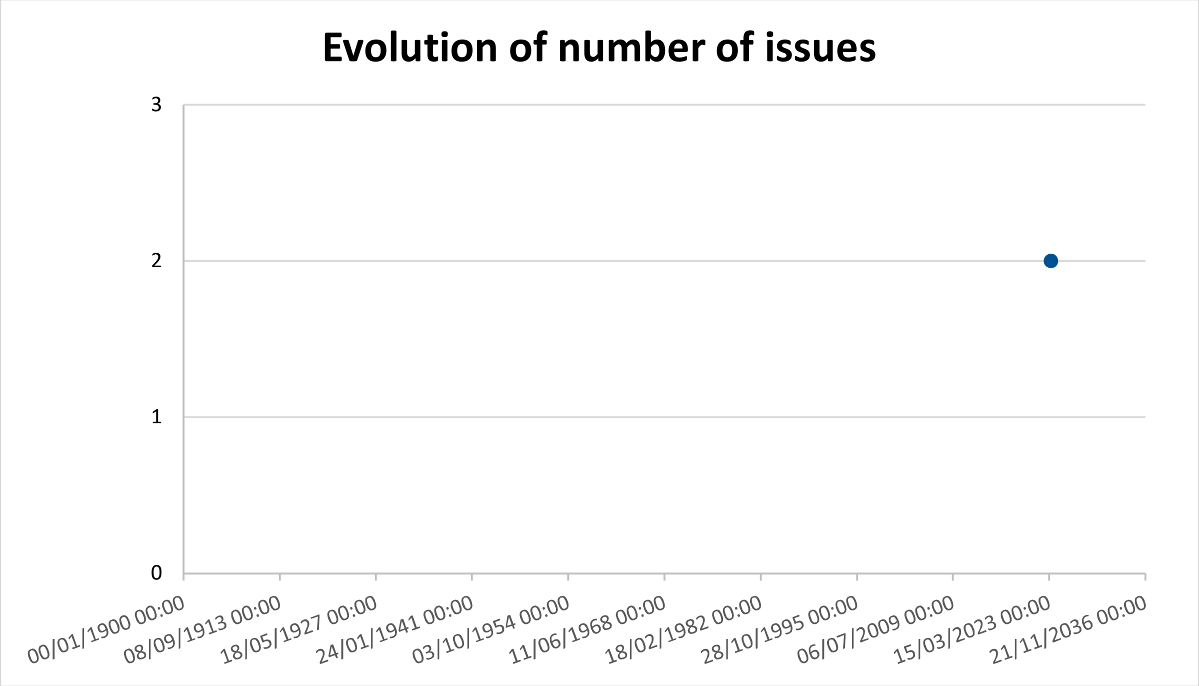
Language	Number
CSS	94
JavaScript	210
HTML	23
Total	327

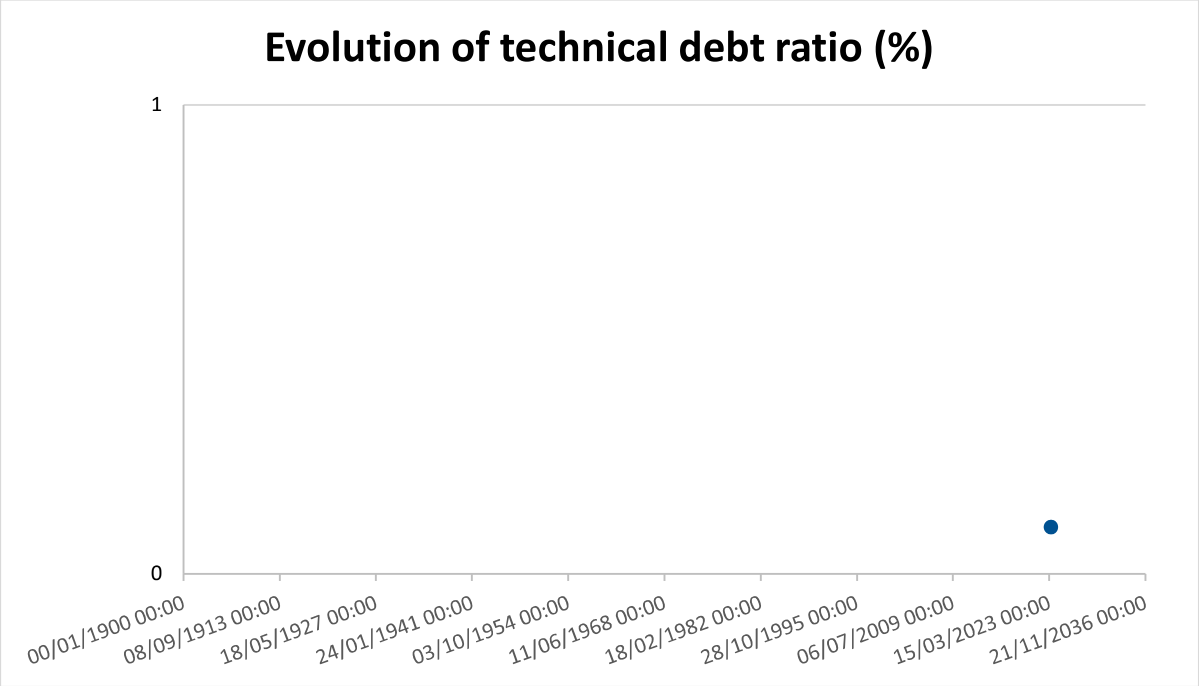
ISSUES

CHARTS






Evolution of number of issues
3
2
1
0
00/01/1900 00:00
08/09/1913 00:00
18/05/1927 00:00
24/01/1941 00:00
03/10/1954 00:00
11/06/1968 00:00
18/02/1982 00:00
28/10/1995 00:00
06/07/2009 00:00
15/03/2023 00:00
21/11/2036 00:00


Evolution of technical debt ratio (%)
1
0
00/01/1900 00:00
08/09/1913 00:00
18/05/1927 00:00
24/01/1941 00:00
03/10/1954 00:00
11/06/1968 00:00
18/02/1982 00:00
28/10/1995 00:00
06/07/2009 00:00
15/03/2023 00:00
21/11/2036 00:00

ISSUES COUNT BY SEVERITY AND TYPE

Type / Severity	INFO	MINOR	MAJOR	CRITICAL	BLOCKER
BUG	0	0	0	0	0
VULNERABILITY	0	0	0	0	0
CODE_SMELL	0	2	0	0	0

ISSUES LIST

Name	Description	Type	Severity	Number
"for of" should be used with Iterables	If you have an iterable, such as an array, set, or list, your best option for looping through its values is the for of syntax. Use a counter, and … well you'll get the right behavior, but your code just isn't as clean or clear. In a browser environment, NodeList and other array-like collections should work by default. If you are using TypeScript and seeing a type error, make sure your configuration is correct. Noncompliant Code Example const arr = [4, 3, 2, 1]; for (let i = 0; i &lt; arr.length; i++) { // Noncompliant console.log(arr[i]); } Compliant Solution const arr = [4, 3, 2, 1]; for (let value of arr) { console.log(value); }	CODE_SMELL	MINOR	2

SECURITY HOTSPOTS

SECURITY HOTSPOTS COUNT BY CATEGORY AND PRIORITY

Category / Priority	LOW	MEDIUM	HIGH
LDAP Injection	0	0	0
Object Injection	0	0	0
Server-Side Request Forgery (SSRF)	0	0	0
XML External Entity (XXE)	0	0	0
Insecure Configuration	0	0	0
XPath Injection	0	0	0
Authentication	0	0	0
Weak Cryptography	0	0	0
Denial of Service (DoS)	0	0	0
Log Injection	0	0	0
Cross-Site Request Forgery (CSRF)	0	0	0
Open Redirect	0	0	0
Permission	0	0	0
SQL Injection	0	0	0
Encryption of Sensitive Data	0	0	0
Traceability	0	0	0
Buffer Overflow	0	0	0
File Manipulation	0	0	0
Code Injection (RCE)	0	0	0

Cross-Site Scripting (XSS)	0	0	0
Command Injection	0	0	0
Path Traversal Injection	0	0	0
HTTP Response Splitting	0	0	0
Others	0	0	0

SECURITY HOTSPOTS LIST