

# EMSAASHTML

Branch : main

VERSION 0.0.0

## Code analysis

By: default

2025-04-24

Analyzed the : 2025-04-17



## CONTENT

Content.....	2
Introduction.....	3
Configuration .....	3
Synthesis .....	4
Analysis Status .....	4
Quality gate status .....	4
Metrics .....	4
Tests.....	4
Detailed technical debt .....	5
Metrics Range .....	6
Volume .....	6
Issues .....	7
Charts .....	7
Issues count by severity and type .....	9
Issues List .....	9
Security Hotspots.....	14
Security hotspots count by category and priority .....	14
Security hotspots List.....	15

## INTRODUCTION

This document contains results of the code analysis of EMSAASHTML.

## CONFIGURATION

- Quality Profiles
  - Names: Sonar way [CSS]; Sonar way [JavaScript]; Sonar way [TypeScript]; Sonar way [HTML];
  - Files: f39a3e8c-bff3-4682-b8f4-fe5c78710032.json; dae334bb-69c0-4b55-be25-06f7e70f7b54.json; a3359080-dc94-4d06-9ee6-5dd4fdc9c082.json; 4341e677-fa7d-45dc-9f59-1c6923806682.json;
- Quality Gate
  - Name: Sonar way
  - File: Sonar way.xml

## SYNTHESIS

### ANALYSIS STATUS

Reliability

Security

Security Review

Maintainability

D

A

E

A

### QUALITY GATE STATUS

Quality Gate Status

Failed

#### Metric

#### Value

Coverage on New Code ERROR (0.0% is less than 80%)

Duplicated Lines (%) on New Code ERROR (16.2% is greater than 3%)

Security Hotspots Reviewed on New Code ERROR (0.0% is less than 100%)

New Issues ERROR (609 is greater than 0)

## METRICS

Coverage

Duplication

Comment density

Median number of lines of code per file

Adherence to coding standard

0.0 %

18.0 %

6.5 %

102.0

96.9 %

## TESTS

Total

Success Rate

Skipped

Errors

Failures

0

0 %

0

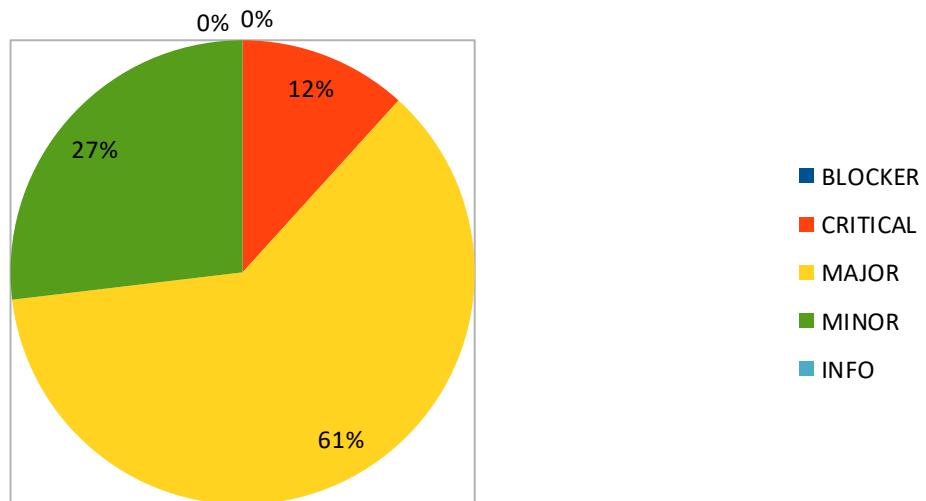
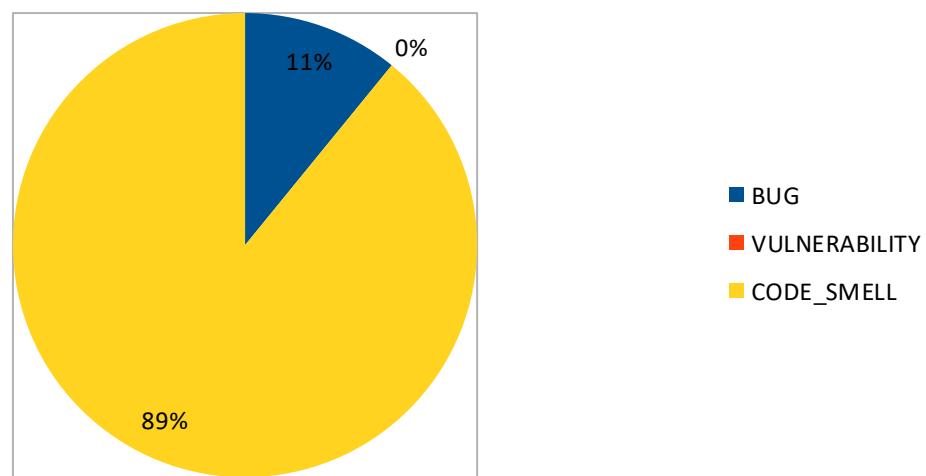
0

0

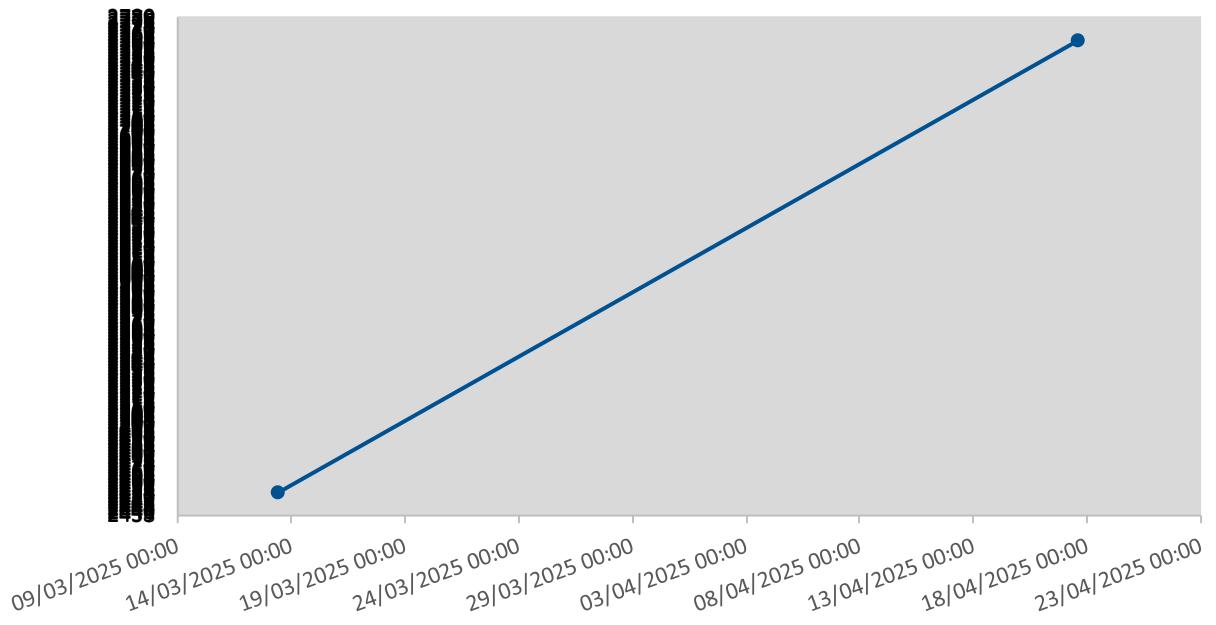
**DETAILED TECHNICAL DEBT**

Reliability	Security	Maintainability	Total
2d 6h 28min	-	28d 2h 1min	<b>31d 0h 29min</b>

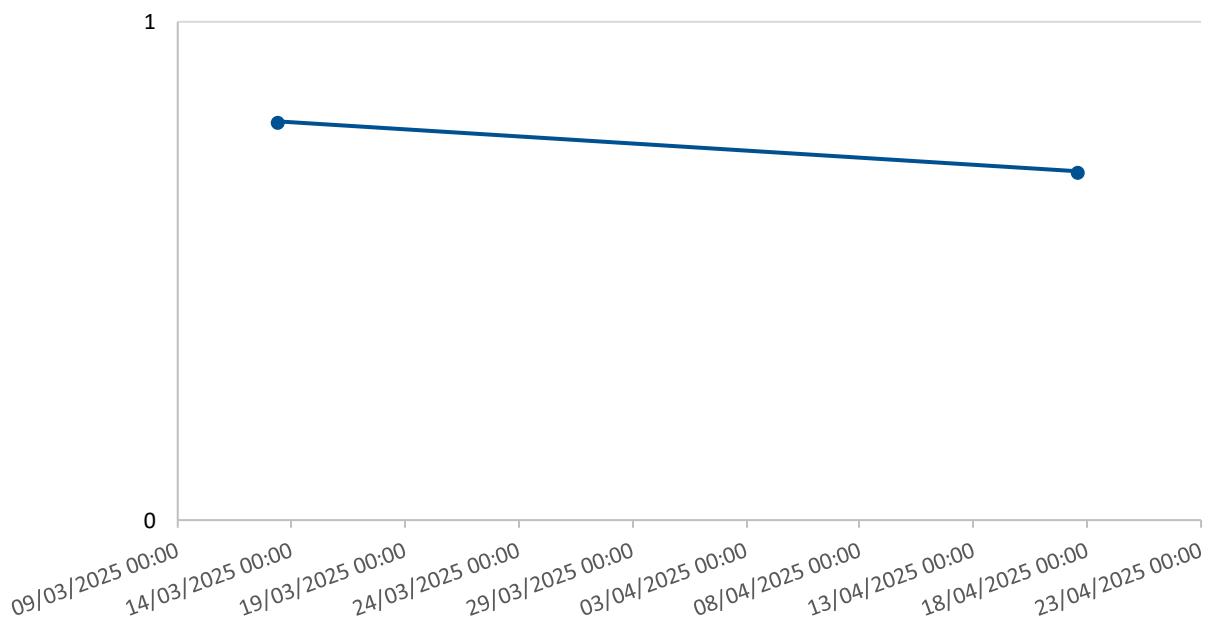
Metrics Range						
	Cyclomatic Complexity	Cognitive Complexity	Lines of code per file	Comment density (%)	Coverage	Duplication (%)
Min	0.0	0.0	0.0	0.0	0.0	0.0
Max	7489.0	3878.0	65354.0	75.0	0.0	98.3
Volume						
Language	Number					
CSS	18040					
JavaScript	1378					
TypeScript	31862					
HTML	14074					
Total	65354					

**ISSUES****CHARTS****Number of issues by severity****Number of issues by type**

## Evolution of number of issues



## Evolution of technical debt ratio (%)



Type / Severity	INFO	MINOR	MAJOR	CRITICAL	BLOCKER
BUG	0	142	145	14	0
VULNERABILITY	0	0	0	0	0
CODE_SMELL	0	601	1552	310	0

ISSUES LIST					
Name	Description	Type	Severity	Number	
Shorthand properties that override related longhand properties should be avoided		BUG	CRITICAL	3	
Selectors should be known		BUG	CRITICAL	11	
"<th>" tags should have "id" or "scope" attributes		BUG	MAJOR	16	
Font declarations should contain at least one generic font family		BUG	MAJOR	32	
Properties should not be duplicated		BUG	MAJOR	24	
Pseudo-element selectors should be valid		BUG	MAJOR	7	
Variables should not be self-assigned		BUG	MAJOR	16	
Identical expressions should not be used on both sides of a binary operator		BUG	MAJOR	1	
"if/else if" chains and "switch" cases should not have the same condition		BUG	MAJOR	1	
Return values from functions without side effects should not be ignored		BUG	MAJOR	12	
All branches in a conditional structure should not have exactly the same implementation		BUG	MAJOR	2	
Optional chaining should not be used if returning "undefined" throws an error		BUG	MAJOR	2	

Promises should not be misused	BUG	MAJOR	23
Binary expressions should not always return the same value	BUG	MAJOR	4
Non-empty statements should change control flow or have at least one side-effect	BUG	MAJOR	5
Mouse events should have corresponding keyboard events	BUG	MINOR	142
Functions should not be nested too deeply	CODE_SMELL	CRITICAL	86
Variables should be declared with "let" or "const"	CODE_SMELL	CRITICAL	177
Functions should not be empty	CODE_SMELL	CRITICAL	11
Functions should not be nested too deeply	CODE_SMELL	CRITICAL	7
Variables should be declared with "let" or "const"	CODE_SMELL	CRITICAL	8
Cognitive Complexity of functions should not be too high	CODE_SMELL	CRITICAL	8
"await" should only be used with promises	CODE_SMELL	CRITICAL	8
Constructors should not contain asynchronous operations	CODE_SMELL	CRITICAL	5
Sections of code should not be commented out	CODE_SMELL	MAJOR	99
Attributes deprecated in HTML5 should not be used	CODE_SMELL	MAJOR	1
DOM elements with ARIA roles should have the required properties	CODE_SMELL	MAJOR	6
Heading elements should have accessible content	CODE_SMELL	MAJOR	1
Label elements should have a text label and an associated control	CODE_SMELL	MAJOR	175
Sections of code should not be commented out	CODE_SMELL	MAJOR	117
Duplicated font names should be removed	CODE_SMELL	MAJOR	2
Empty blocks should be removed	CODE_SMELL	MAJOR	2

Selectors should not be duplicated	CODE_SMELL	MAJOR	101
CSS files should not be empty	CODE_SMELL	MAJOR	45
Functions should not have too many parameters	CODE_SMELL	MAJOR	1
Assignments should not be made from within sub-expressions	CODE_SMELL	MAJOR	24
Sections of code should not be commented out	CODE_SMELL	MAJOR	2
Unused assignments should be removed	CODE_SMELL	MAJOR	1
Multiline blocks should be enclosed in curly braces	CODE_SMELL	MAJOR	3
Ternary operators should not be nested	CODE_SMELL	MAJOR	1
Unnecessary character escapes should be removed	CODE_SMELL	MAJOR	3
Calls to ".call()" and ".apply()" methods should not be redundant	CODE_SMELL	MAJOR	2
Comma operator should not be used	CODE_SMELL	MAJOR	2
Functions should not have too many parameters	CODE_SMELL	MAJOR	7
Nested blocks of code should not be left empty	CODE_SMELL	MAJOR	14
Assignments should not be made from within sub-expressions	CODE_SMELL	MAJOR	11
Sections of code should not be commented out	CODE_SMELL	MAJOR	236
Unused assignments should be removed	CODE_SMELL	MAJOR	68
Two branches in a conditional structure should not have exactly the same implementation	CODE_SMELL	MAJOR	6
Fields that are only assigned in the constructor should be "readonly"	CODE_SMELL	MAJOR	551
Ternary operators should not be nested	CODE_SMELL	MAJOR	1
Collection contents should be used	CODE_SMELL	MAJOR	1

Functions should not have identical implementations	CODE_SMELL	MAJOR	4
Assignments should not be redundant	CODE_SMELL	MAJOR	2
Unnecessary character escapes should be removed	CODE_SMELL	MAJOR	1
Optional chaining should be preferred	CODE_SMELL	MAJOR	30
Enum members should not mix value types	CODE_SMELL	MAJOR	1
If statements should not be the only statement in else blocks	CODE_SMELL	MAJOR	5
Literals should not be used for promise rejection	CODE_SMELL	MAJOR	15
Comma operator should not be used	CODE_SMELL	MAJOR	11
Image, area and button with image elements should have an "alt" attribute	CODE_SMELL	MINOR	49
Extra semicolons should be removed	CODE_SMELL	MINOR	11
Duplicate imports should be removed	CODE_SMELL	MINOR	3
Unused local variables and functions should be removed	CODE_SMELL	MINOR	3
"for of" should be used with Iterables	CODE_SMELL	MINOR	2
Regular expression quantifiers and character classes should be used concisely	CODE_SMELL	MINOR	1
Ternary operator should not be used instead of simpler alternatives	CODE_SMELL	MINOR	1
Use Object.hasOwnProperty static method instead of hasOwnProperty	CODE_SMELL	MINOR	1
Class names should comply with a naming convention	CODE_SMELL	MINOR	8
Boolean literals should not be used in comparisons	CODE_SMELL	MINOR	8
Unnecessary imports should be removed	CODE_SMELL	MINOR	97

Wrapper objects should not be used for primitive types	CODE_SMELL	MINOR	1
Deprecated APIs should not be used	CODE_SMELL	MINOR	263
"catch" clauses should do more than rethrow	CODE_SMELL	MINOR	12
Jump statements should not be redundant	CODE_SMELL	MINOR	3
Imports from the same module should be merged	CODE_SMELL	MINOR	8
"for of" should be used with Iterables	CODE_SMELL	MINOR	4
Redundant casts and non-null assertions should be avoided	CODE_SMELL	MINOR	1
Regular expression quantifiers and character classes should be used concisely	CODE_SMELL	MINOR	54
Type constituents of unions and intersections should not be redundant	CODE_SMELL	MINOR	1
"RegExp.exec()" should be preferred over "String.match()"	CODE_SMELL	MINOR	1
Nullish coalescing should be preferred	CODE_SMELL	MINOR	33
Ternary operator should not be used instead of simpler alternatives	CODE_SMELL	MINOR	36

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	1
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	4	0	0
Traceability	0	0	0
Buffer Overflow	0	0	0
File Manipulation	0	0	0
Code Injection (RCE)	0	0	0

## EMSAASHTML

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	2	0	0

SECURITY HOTSPOTS LIST					
Category	Name	Priority	Severity	Count	
Others	Using intrusive permissions is security-sensitive	LOW	MAJOR	1	
Authentication	Hard-coded passwords are security-sensitive	HIGH	BLOCKER	1	
Others	Using remote artifacts without integrity checks is security-sensitive	LOW	MINOR	1	
Encryption of Sensitive Data	Using clear-text protocols is security-sensitive	LOW	CRITICAL	4	