Measures

Reliability

Overview

On new code

Bugs — 0

Rating — A

Remediation Effort — 0

Overall

Bugs — 1

Rating — B

Remediation Effort — 2min

Security

Overview

On new code

Vulnerabilities — 0

Rating — A

Remediation Effort — 0

Overall

Vulnerabilities — 0

Rating — A

Remediation Effort — 0

Security Review

On new code

Security Hotspots — 0

Rating — A

Overall

Security Hotspots — 1

Rating — E

Security Hotspots Reviewed — 0.0%

⌄ Maintainability ⊘

Overview   ᕲ

On new code

Code Smells	0
Debt	0
Debt Ratio	0.0%
Rating	**A**

Overall

Code Smells	9
Debt	44min
Debt Ratio	0.3%
Rating	**A**
Effort to Reach A	0

⌄ Coverage

Overview   ᕲ

On new code

Lines to Cover	0
Uncovered Lines	0
Conditions to Cover	0
Uncovered Conditions	0

Overall

Coverage	0.0%
Lines to Cover	203
Uncovered Lines	203
Line Coverage	0.0%

⌄ Duplications

Overview   ⌗

On new code

Duplicated Lines	0
Duplicated Blocks	0

Overall

Density	0.0%
Duplicated Lines	0
Duplicated Blocks	0
Duplicated Files	0

⌄ Size

New Lines	0
Lines of Code	546
Lines	9,833
Statements	203
Functions	85
Classes	1
Files	26
Comment Lines	13
Comments (%)	2.3%

⌄ Complexity ⊙

Cyclomatic Complexity	100
Cognitive Complexity	13

✓ Issues

New Issues	0
Issues	10
Open Issues	10
Reopened Issues	0
Confirmed Issues	0
False Positive Issues	0
Won't Fix Issues	0

Overall

0 🐞 New Bugs		Reliability	A
0 🔒 New Vulnerabilities		Security	A
0 🛡 New Security Hotspots ❓	— Reviewed	Security Review	A
0 Added Debt	0 ⊛ New Code Smells	Maintainability	A

Security

Make sure the regex used here, which is vulnerable to super-linear runtime due to backtracking, cannot lead to denial of service.

Using slow regular expressions is security-sensitive   typescript:S5852

Add Comment   Open in IDE   % Get Permalink

Category	**Denial of Service (DoS)**
Review priority	MEDIUM
Assignee	**Not assigned** ✏

Status: To review

This Security Hotspot needs to be reviewed to assess whether the code poses a risk.

Change status ▾

Server/controllers/chat/chatMessageVerifiers.ts

```
1    export const isHTML = function(message:string) : boolean {
2        return /((<\w*)((\s\/\>)|(.*\<\/\w*\>))/i.test(message);
3    }
4
5    export const isEncodedURL = function(message:string) : boolean {
6        return /%[0-9A-F]/i.test(message);
7    }
```