

commons-jexl

 Filter 

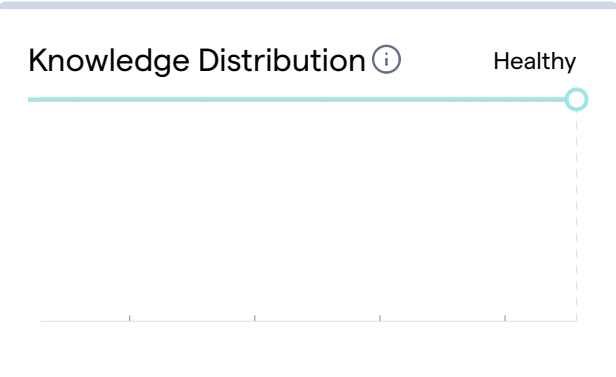
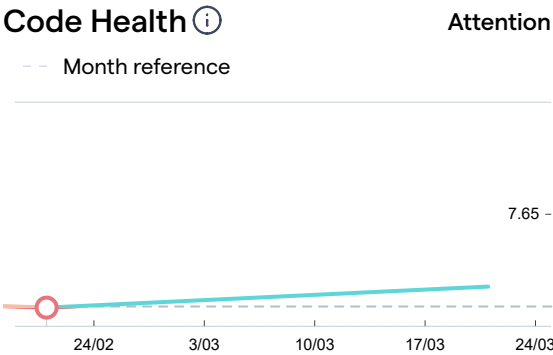

1W

1M

1Y

 Lines of code
39,689 (/2/analyses/2/scope/analysis-data/file-content)

Programming languages



Team-Code Alignment ⓘ Not available

Additional configuration steps are required before this factor can be used.

Define teams (/developer-configurations/for-project/2/developers)

Deliver

Additional configuration steps are required before this factor can be used.

Configure integrations

ⓘ What is Code Health?



Hotspot Code Health ⓘ

Problematic

Hotspots are the files with most development activity. Even a minor amount of technical debt in a hotspot will become expensive due to the high development activity.

View hotspots
(/2/analyses/2/code/biomarkers?)

Average Code Health ⓘ

Problematic

The weighted average health of all the files in the codebase. This metric indicates how deep any potential code health issues go.

Explore codebase
(/2/analyses/2/code/hotspots/system-map?max-code-health=10&min-change-freq=default#codehealth)

Worst Performer ⓘ

Unhealthy

The lowest code health score measured in the codebase. Points out long-term risks that you need to be aware of if the low performer is worked on.

View worst performers
(/2/analyses/2/code/hotspots/system-map?max-code-health=2.2561789&min-change-freq=0#codehealth)



Latest improvements

No improvements available



Active risks & improvement opportunities



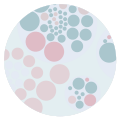
4 Hotspots decline in Code Health

Hotspots are degrading in code health, which makes the system more expensive to maintain.

[View list \(/2/analyses/2/warnings/code-health-degradation\)](/2/analyses/2/warnings/code-health-degradation)



Code health focus areas



Interactive hotspot map

Visualize hotspots, code health, defects, etc. and filter by team and area.

Red code	12.8%
Yellow code	50.6%
Green code	36.6%

[View hotspots \(/2/analyses/2/code/hotspots/system-map?max-code-health=10&min-change-freq=default#codehealth\)](/2/analyses/2/code/hotspots/system-map?max-code-health=10&min-change-freq=default#codehealth)



File-level hotspots

Prioritizing improvements to these files will result in the greatest immediate benefit in terms of technical debt.

Hotspots

7.4%

Development effort in hotspots

30.2%

[View hotspots \(/2/analyses/2/code/biomarkers?\)](/2/analyses/2/code/biomarkers?)



Architectural code health

Monitor the code health of architectural components.

You need to **configure (/projects/2/configuration#architecture)** architectural components and rerun the analysis to see this data.

[Configure components \(/projects/2/configuration#architecture\)](/projects/2/configuration#architecture)



Quality gates

Track the impact of CodeScene's Pull Request integration. Numbers are from current month.

PR checks performed	0
Negative findings	0
Findings fixed	0
Findings ignored	0
Findings suppressed	0

[View PR statistics \(/2/analyses/2/project-management/delta/statistics\)](/2/analyses/2/project-management/delta/statistics)



Planned vs. unplanned work

This is time spent on work that was not part of the original scope.

Couldn't discover completed tasks. You need to configure a Work Done Transition name in your PM configuration.

[View unplanned work \(/2/analyses/2/project-management/hotspots/system-level\)](/2/analyses/2/project-management/hotspots/system-level)



Change coupling

Identify implicit dependencies in your architecture.

File-level

By commits	89 entities
Across commits	67 entities
Between repositories	0 entities

View coupled entities (/2/analyses/2/code/temporal-coupling/by-commits)