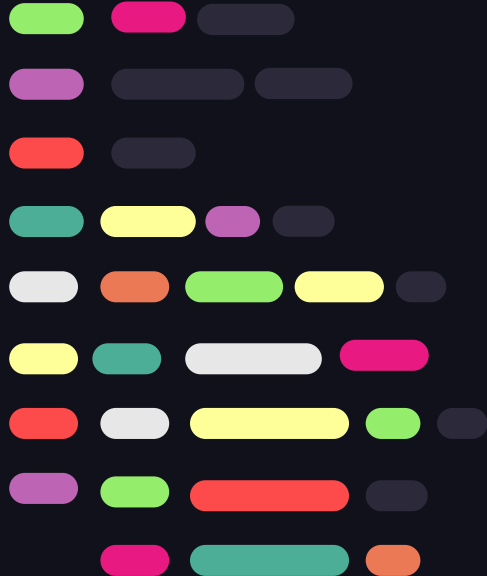




Apache Commons Validator Library

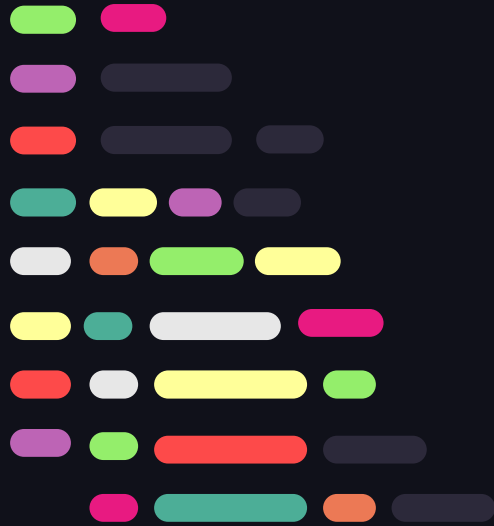
Software Dependability Analysis

Katia Melanie Perchet





Content

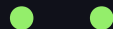


- 01 Project Goals
- 02 Apache Commons Validator
- 03 Analysis SonarCloud
- 04 Code Coverage
- 05 Mutation Testing
- 06 Benchmarking Tools
- 07 Automated Tests
- 08 Software Vulnerabilities

Project Goals



It is aim to provide a comprehensive evaluation of the Apache Commons Validator library's dependability, addressing aspects of code quality, test coverage, security, and performance.



Apache Commons Validator



Is an open-source Java library which provides the ability to verify the integrity of incoming data, both client-side and server-side.



SonarCloud Analysis

{ Issues

After a thorough analysis of the project, the report revealed an extensible list of issues.





Issues Found



HIGH

- Clone methods
- Serializable or transient variables
- Empty methods
- Absence of assertions in tests



MEDIUM

- Extensive ReGex
- Nested try-catch - Inappropriate constructor's visibility
- Wrong of method arguments



LOW

- Deprecated and commented code.
- Singleton Pattern
- Boolean literals
- Public modifiers in Tests





Issues Fixed



Addressed the majority of issues, of which four of the bugs were fixed and five hundred and fifty-one code smells

Katia Perchet > Apache Commons Validator > master ☒

Summary Issues Security Hotspots Measures Code Activity

7.8k Lines of Code [?](#) Version 1.8.1-SNAPSHOT Last analysis 1 hour ago [79233674](#) variables still in use

☒ Quality Gate [?](#) **Passed** [New Code](#) [Overall Code](#)

⚙ Reliability 1 Bugs ? C	⚙ Maintainability 93 Code Smells ? A
🔒 Security 0 Vulnerabilities ? A	🔒 Security Review 5 Security Hotspots ? 0.0% Reviewed E



Code Coverage



JaCoCo

Report

It generated comprehensive report that points to specific sections of the code exercised by the tests.

Apache Commons Validator													
Apache Commons Validator													
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes	
org.apache.commons.validator	<div><div></div></div>	64%	<div><div></div></div>	60%	344	719	513	1,502	123	337	2	26	
org.apache.commons.validator.routines	<div><div></div></div>	97%	<div><div></div></div>	86%	105	787	101	1,273	9	363	0	34	
org.apache.commons.validator.util	<div><div></div></div>	39%	<div><div></div></div>	32%	24	42	41	73	8	22	0	2	
org.apache.commons.validator.routines.checkdigit	<div><div></div></div>	96%	<div><div></div></div>	96%	9	133	10	221	4	65	0	14	
Total	2,859 of 21,182	86%	442 of 1,762	74%	482	1,681	665	3,069	144	787	2	76	



Mutation Testing



PiTest

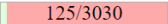
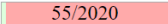
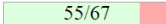
It aims to enhance test adequacy and pinpoint defects within the codebase by revealing weaknesses in test coverage.

Report

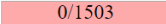
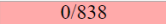
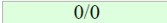
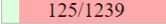
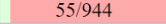
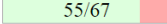
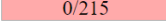
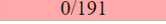
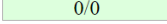
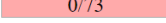
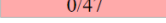
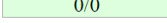


Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage	Mutation Coverage	Test Strength
61	4% 	3% 	82% 

Breakdown by Package

Name	Number of Classes	Line Coverage	Mutation Coverage	Test Strength
org.apache.commons.validator	19	0% 	0% 	100% 
org.apache.commons.validator.routines	27	10% 	6% 	82% 
org.apache.commons.validator.routines.checkdigit	13	0% 	0% 	100% 
org.apache.commons.validator.util	2	0% 	0% 	100% 



Benchmarking Tools

Java Microbenchmarking Harness (JMH)



Tool designed for the precise and dependable benchmarking of Java code, facilitating accurate measurement and analysis within a controlled and well-defined framework.

Benchmark	Mode	Cnt	Score	Error	Units
BenchmarkRunner.testIsValid1Benchmark	thrpt	25	336728.518 ±	12381.527	ops/s
BenchmarkRunner.testIsValid2Benchmark	thrpt	25	59810.182 ±	3980.859	ops/s
BenchmarkRunner.testIsValidAuth1Benchmark	thrpt	25	402688.222 ±	14788.048	ops/s
BenchmarkRunner.testIsValidAuth2Benchmark	thrpt	25	60322.020 ±	2688.195	ops/s

Process finished with exit code 0

Automated Testing Generation

EvoSuite



Generated test suites with two primary goals: maximizing a specified coverage criterion and minimizing the suite's size in terms of the number of tests and statements.

Results

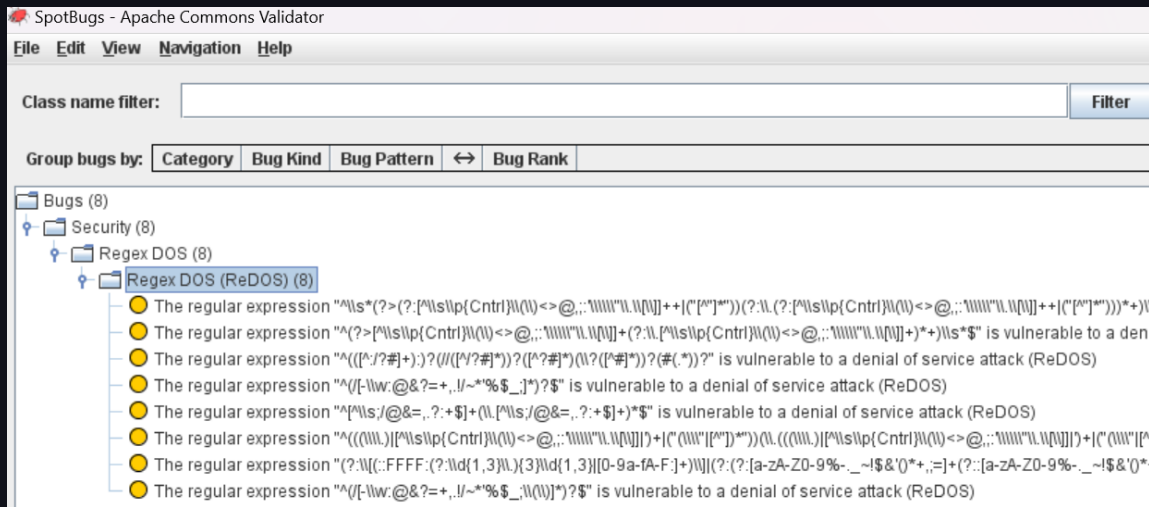
This process resulted in the creation of 129 new tests for enhanced code coverage. Each class was equipped with two types of tests: the primary test and accompanying scaffolding classes.



Software Vulnerabilities

FindSecBugs

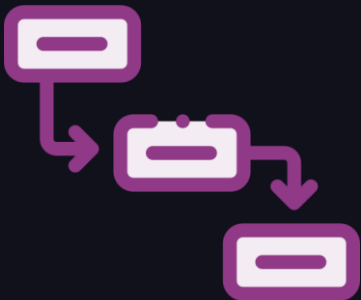
Automated security analysis tool, leveraging its capabilities to scrutinize Java code comprehensively.



Software Vulnerabilities

OWASP Dependency Check

Helped to identify and mitigate security risks associated with third-party dependencies.



Summary

Display: [Showing All Dependencies \(click to show less\)](#)

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
commons-beanutils-1.9.4.jar	cpe:2.3:a:apache:commons_beanutils:1.9.4:*:*:*:*:*	pkg:maven/commons-beanutils/commons-beanutils@1.9.4		0	Highest	168
commons-collections-3.2.2.jar	cpe:2.3:a:apache:commons_collections:3.2.2:*:*:*:*	pkg:maven/commons-collections/commons-collections@3.2.2		0	Highest	84
commons-digester-2.1.jar		pkg:maven/commons-digester/commons-digester@2.1		0		98
commons-logging-1.3.0.jar		pkg:maven/commons-logging/commons-logging@1.3.0		0		129
commons-math3-3.2.jar		pkg:maven/org.apache.commons/commons-math3@3.2		0		125
jmh-core-1.36.jar		pkg:maven/org.openjdk.jmh/jmh-core@1.36		0		27
jmh-generator-annprocess-1.36.jar		pkg:maven/org.openjdk.jmh/jmh-generator-annprocess@1.36		0		25
jopt-simple-5.0.4.jar		pkg:maven/net.sf.jopt-simple/jopt-simple@5.0.4		0		23
pitest-1.15.6.jar		pkg:maven/org.pitest/pitest@1.15.6		0		23

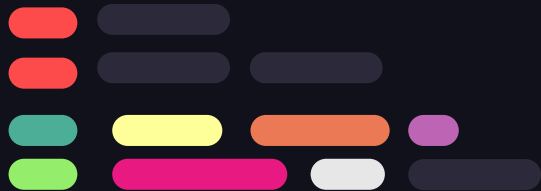


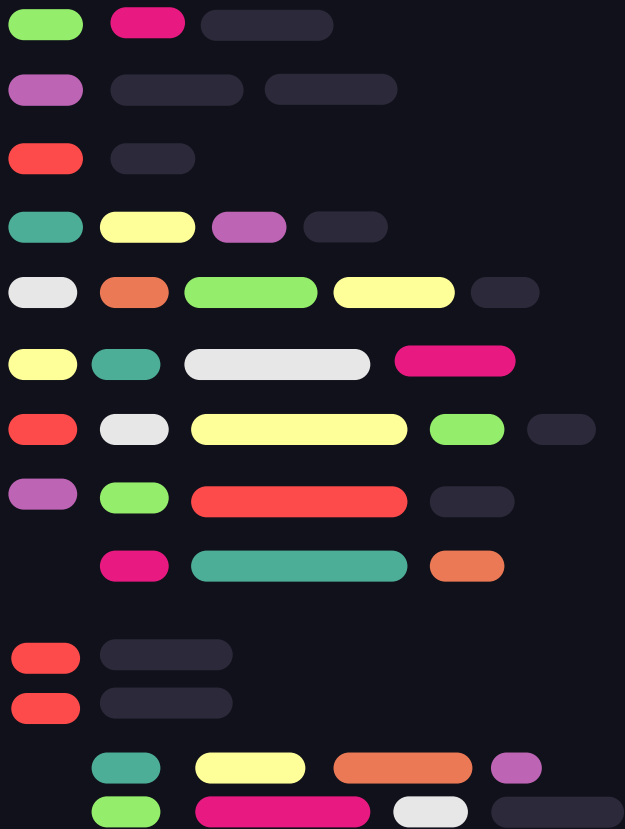
Conclusion



The implementation of these tools has significantly elevated the Apache Commons Validator project's dependability, significantly enhanced its overall quality, reliability, and security.

Adopting these practices as part of a continuous improvement strategy will fortify the existing project and promote a culture of excellence and resilience in software engineering.





{ ..



Thank
you.

} ..