Apache Commons
Validator

Dependability Analysis

Nicolò Delogu - Mat. 0522501556
Dario Mazza - Mat. 0522501553

Report

Source code

INTRODUCTION

Apache Commons Validator is a widely-used library for data validation in Java applications.

Our objective is to assess the dependability of this project through a series of structured steps.

Report

Source code

TABLE OF CONTENTS

TABLE OF CONTENTS

01

CI/CD

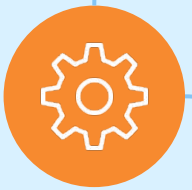
CI/CD

Workflows

Yaml files definition

Execution

Upon commit onto a branch



Github Actions

Monitoring

CI/CD

Workflows

Yaml files definition

Pre-existing workflows

Maven™

What we added

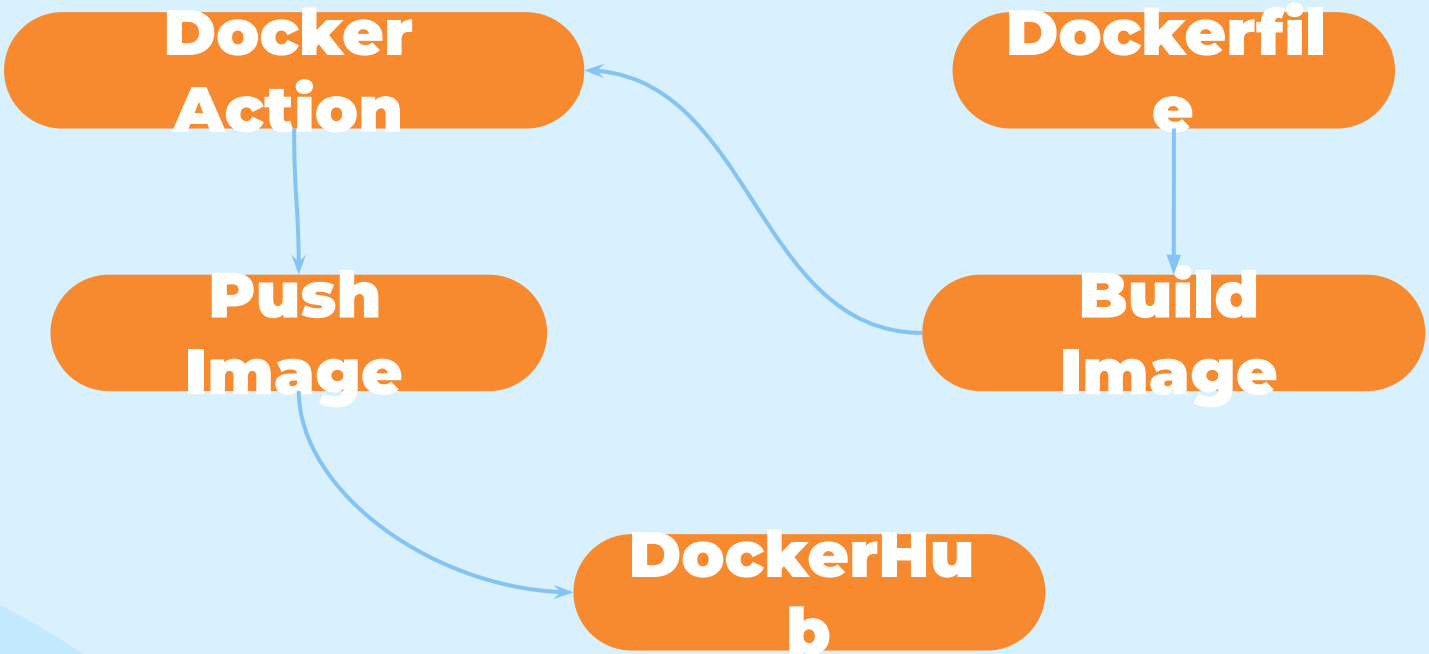
docker

02

Web App Deployment

Web App Deployment

Workflow

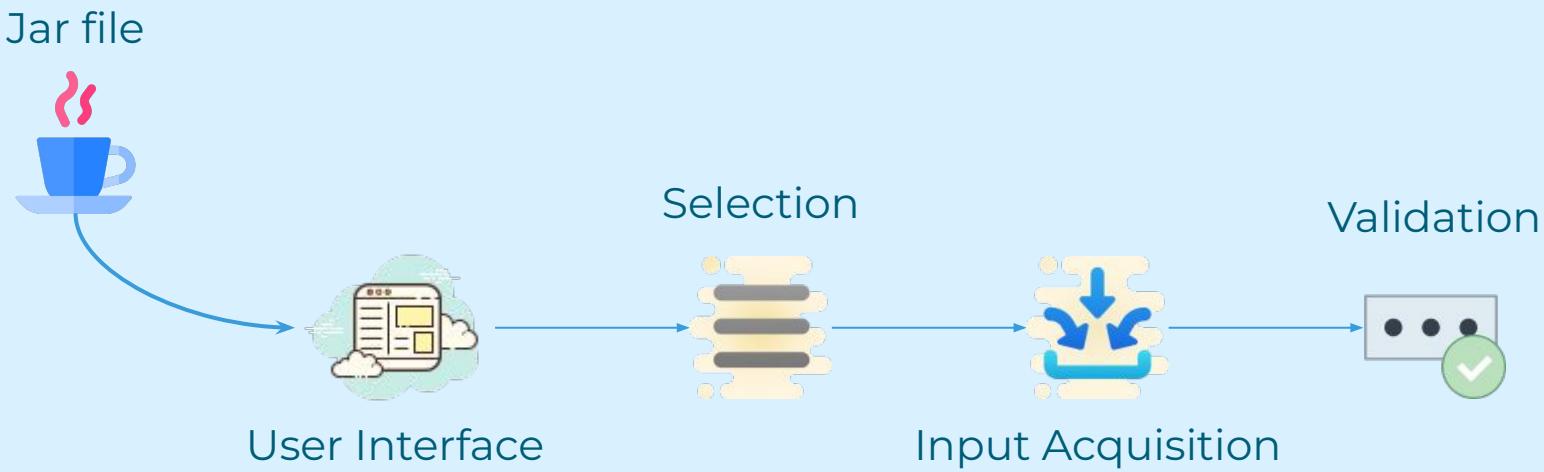


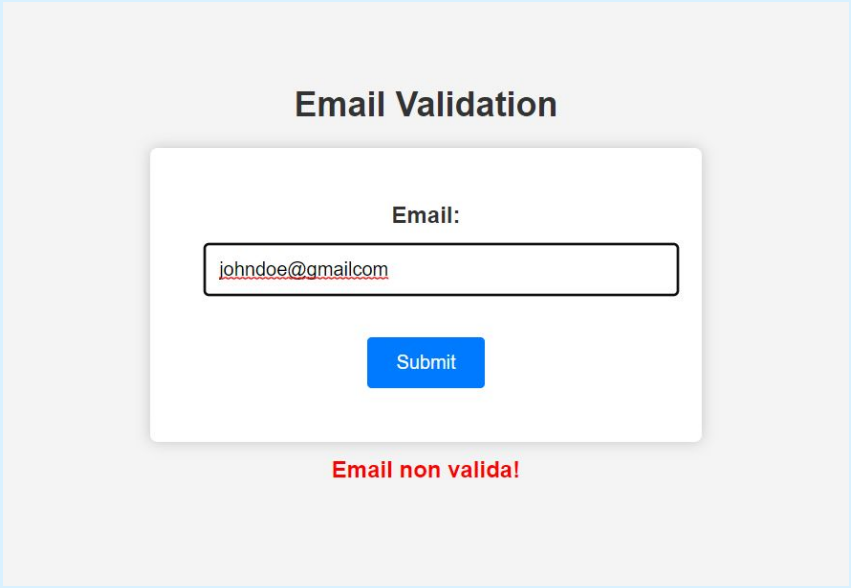
Web App Deployment

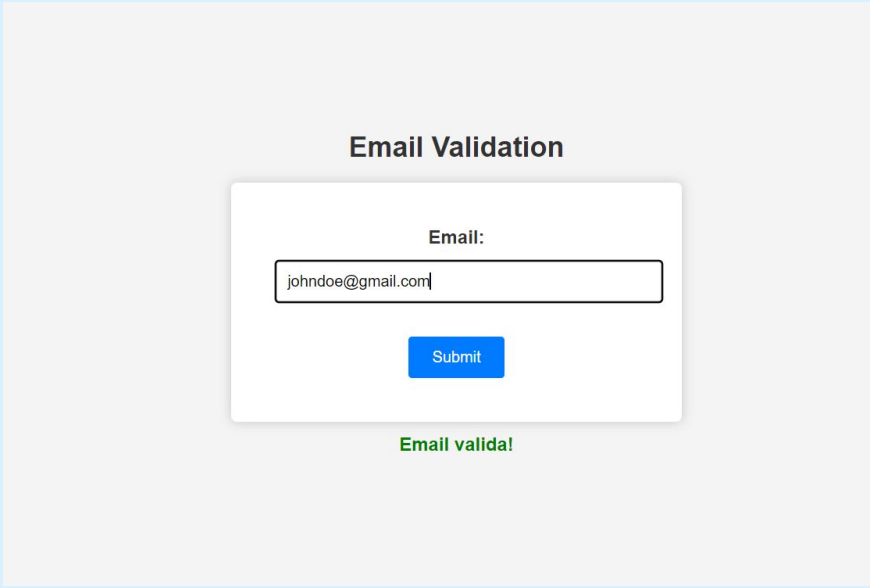
Dockerfile

```
FROM maven:3.9.2-eclipse-temurin-17 as base
WORKDIR /app
RUN groupadd -r myuser && useradd --create-home -r -g myuser myuser
RUN usermod -u 1024 myuser; groupmod -g 1024 myuser;

COPY commons-validator-1.8-SNAPSHOT.jar /app/commons-validator-1.8-SNAPSHOT.jar
COPY ./pom.xml ./

RUN mvn dependency:resolve
RUN mkdir target RUN chown myuser target
COPY ./src ./src
USER myuser
RUN mvn package

FROM eclipse-temurin:17-jre-jammy as production
EXPOSE 8080/tcp
COPY--from=base/app/target/validator-web-app-0.0.1-SNAPSHOT.jar/commons-validator-
webapp.jar
CMD ["java", "-jar", "/commons-validator-webapp.jar"]
```

Web App Deployment



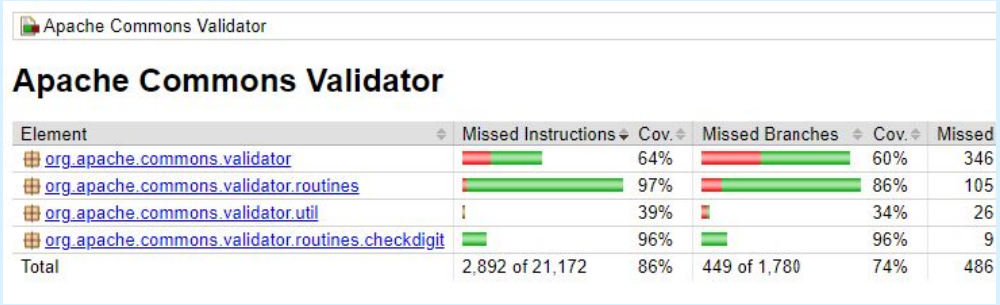

Email Validation
Email:
johndoe@gmailcom
Submit
Email non valida!


Email Validation
Email:
johndoe@gmail.com
Submit
Email valida!

03

Coverage

Coverage

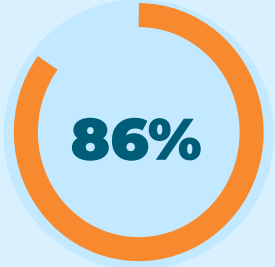

Apache Commons Validator

Apache Commons Validator

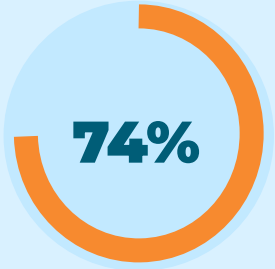
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed
org.apache.commons.validator		64%		60%	346
org.apache.commons.validator.routines		97%		86%	105
org.apache.commons.validator.util		39%		34%	26
org.apache.commons.validator.routines.checkdigit		96%		96%	9
Total	2,892 of 21,172	86%	449 of 1,780	74%	486

Strong Coverage

Instruction Coverage



Branch Coverage



04

Mutation Testing

Mutation Testing

Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
61	72%	2208/3077	68%	1380/2035	89%	1380/1546

Breakdown by Package

Name	Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
org.apache.commons.validator	19	65%	992/1523	55%	459/841	79%	459/581
org.apache.commons.validator.routines	27	77%	968/1254	75%	717/950	96%	717/750
org.apache.commons.validator.routines.checkdigit	13	97%	208/214	94%	180/191	96%	180/188
org.apache.commons.validator.util	2	47%	40/86	45%	24/53	89%	24/27

Report generated by PIT 1.14.1

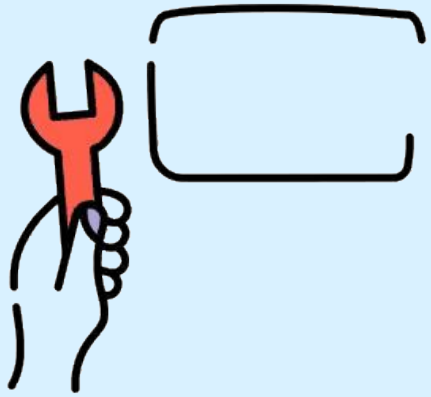
Fairly Robust

Mutation Coverage

Test Strength

05

Refactoring



Refactoring

Code Smells

Multiple if statements instead of switch statements

Test cases devoid of assertions

Replacement of clone methods

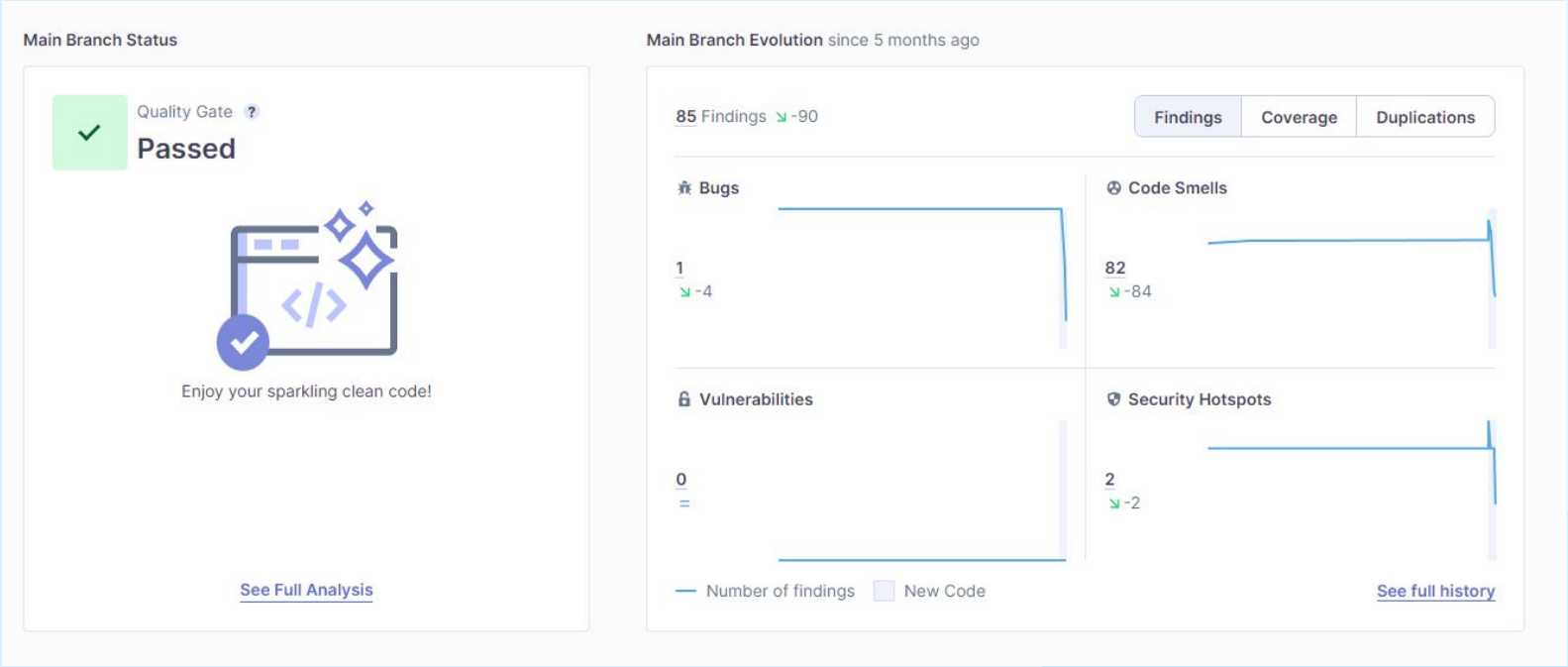
Return statements that need simplification

Steer clear of global variables

Refactoring

SonarCloud Overview

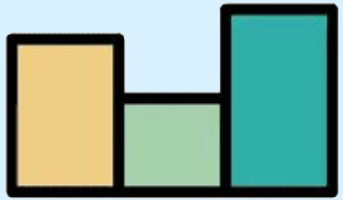
From 166 to 82 code smells



Refactoring

Eco-Design

- Use of arrays within for each loops → More efficient while loops
- Invocation of methods within the evaluation of for loop conditions → Method calls outside of for loops conditions, less resources utilization
- The preference for i++ over ++i in for loops → Switch from i++ to ++i for better performance



06

Benchmarking

BenchMarking

PreRefactoring

PostRefactoring

Comparable Performances

Better maintainability and readability

BenchMarking

EmailValidator

PreRefactoring

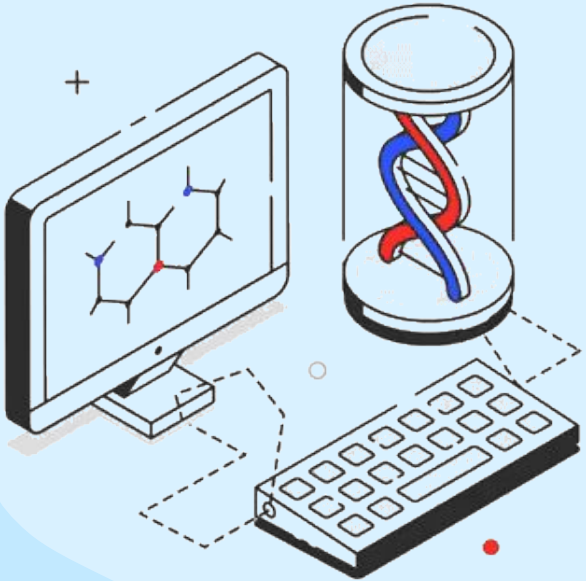
Benchmark	Mode	Cnt	Score	Error	Units
EmailValidatorBenchmark.validateLongInvalidEmail	thrpt	5	224815,318 ±	3336,366	ops/s
EmailValidatorBenchmark.validateLongValidEmail	thrpt	5	215296,812 ±	5484,094	ops/s
EmailValidatorBenchmark.validateMultipleInvalidEmails	thrpt	5	1042,914 ±	9,296	ops/s
EmailValidatorBenchmark.validateMultipleValidEmails	thrpt	5	581,725 ±	6,656	ops/s
EmailValidatorBenchmark.validateShortInvalidEmail	thrpt	5	1855303,076 ±	40905,942	ops/s
EmailValidatorBenchmark.validateShortValidEmail	thrpt	5	1032211,155 ±	12495,359	ops/s
EmailValidatorBenchmark.validateSingleInvalidEmail	thrpt	5	1322249,406 ±	40493,070	ops/s
EmailValidatorBenchmark.validateSingleValidEmail	thrpt	5	733021,905 ±	15371,851	ops/s
EmailValidatorBenchmark.validateVeryLongInvalidEmail	thrpt	5	240206,400 ±	2666,202	ops/s

BenchMarking

EmailValidator

PostRefactoring

Benchmark	Mode	Cnt	Score	Error	Units
EmailValidatorBenchmark.validateLongInvalidEmail	thrpt	5	226023,649 ±	5185,551	ops/s
EmailValidatorBenchmark.validateLongValidEmail	thrpt	5	220097,621 ±	6768,077	ops/s
EmailValidatorBenchmark.validateMultipleInvalidEmails	thrpt	5	1018,256 ±	18,111	ops/s
EmailValidatorBenchmark.validateMultipleValidEmails	thrpt	5	594,920 ±	16,374	ops/s
EmailValidatorBenchmark.validateShortInvalidEmail	thrpt	5	1825389,316 ±	30233,167	ops/s
EmailValidatorBenchmark.validateShortValidEmail	thrpt	5	1025217,280 ±	22905,307	ops/s
EmailValidatorBenchmark.validateSingleInvalidEmail	thrpt	5	1359427,624 ±	23068,646	ops/s
EmailValidatorBenchmark.validateSingleValidEmail	thrpt	5	740675,074 ±	11576,712	ops/s
EmailValidatorBenchmark.validateVeryLongInvalidEmail	thrpt	5	239932,833 ±	4173,851	ops/s



07

Test Suite Generation

Test Suite Generation

Tools

EvoSuite and **Randoop** are both tools used in the field of software testing, particularly for automatic test case generation in Java applications.

Randoop

Generates random sequences of method calls and constructor invocations

Evosuite

Genetic algorithms and search-based techniques

Test Suite Generation

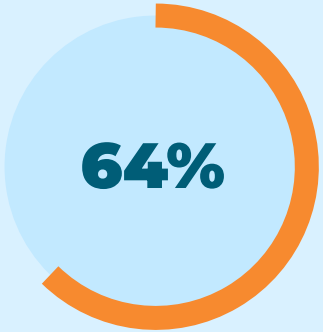
Validator package

org.apache.commons.validator

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed
Field		56%		67%	54
ValidatorAction		59%		55%	42
FormSet		46%		36%	22
GenericValidator		41%		35%	46
EmailValidator		13%		0%	21
ValidatorResources		85%		79%	19
GenericTypeValidator		80%		52%	48
Msg		0%		n/a	11
Form		77%		76%	10
Var		31%		0%	4
Arg		30%		n/a	5
DateValidator		0%		0%	12
Validator		58%		50%	16
ValidatorResources.new Rule() {...}		8%		0%	2
FormSetFactory		69%		62%	3
UrlValidator		93%		89%	11
ValidatorResult		76%		50%	5
ValidatorResults		86%		80%	5
ValidatorResult.ResultStatus		53%		n/a	3
ValidatorException		57%		n/a	1
CreditCardValidator		98%		96%	1
CreditCardValidator.Discover		93%		50%	2
CreditCardValidator.Mastercard		100%		75%	1
CreditCardValidator.Amex		100%		75%	1
CreditCardValidator.Visa		100%		83%	1
ISBNValidator		100%		n/a	0
Total	2.243 of 6.235	64%	306 of 772	60%	346

PreGen

Instruction Coverage



Branch Coverage

Test Suite Generation

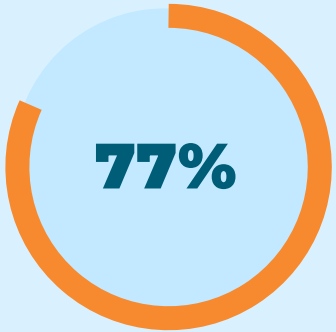
Validator package

org.apache.commons.validator

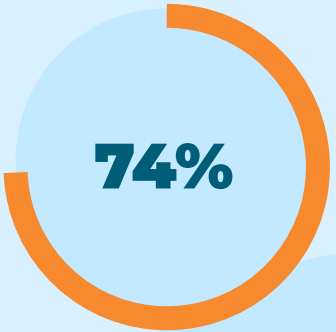
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed
Field		56%		67%	54
ValidatorAction		59%		55%	42
ValidatorResources		85%		79%	19
GenericTypeValidator		81%		52%	48
EmailValidator		68%		46%	11
Validator		64%		64%	14
ValidatorResources.new Rule() {...}		8%		0%	2
Form		85%		79%	8
FormSetFactory		69%		62%	3
UrlValidator		93%		89%	11
ValidatorResult		76%		50%	5
ValidatorResults		86%		80%	5
ValidatorResult.ResultStatus		53%		n/a	3
Var		94%		100%	0
Arg		93%		n/a	0
Msg		92%		n/a	0
DateValidator		90%		100%	0
FormSet		98%		97%	1
CreditCardValidator		98%		96%	1
CreditCardValidator.Discover		93%		50%	2
GenericValidator		100%		100%	0
CreditCardValidator.Mastercard		100%		75%	1
CreditCardValidator.Amex		100%		75%	1
CreditCardValidator.Visa		100%		83%	1
ISBNValidator		100%		n/a	0
ValidatorException		100%		n/a	0
Total	1.416 of 6.235	77%	197 of 772	74%	232

PostGen

Instruction Coverage



Branch Coverage



08

Vulnerabilities

Vulnerabilities

Tools

FindSecBugs

Static code analysis tool that is used to identify potential software defects or vulnerabilities in Java programs.

OwaspDPC

Open-source security tool designed to help identify and reduce security vulnerabilities in an application's dependencies or third-party libraries.

Vulnerabilities

OwaspDPC

- Bouncy Castle versions prior to 1.74
- H2 Database Engine up to version 2.1.214
- Jackson-databind library up to version 2.15.2
- Maven core 3.8.0 and Maven settings 3.0
- Maven share utils 3.1.0

FindSecBugs

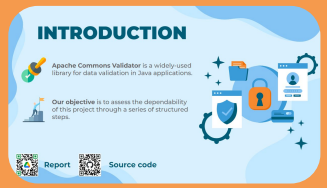
↓

Manual Review

↓

False positives

Thanks!



01 CI/CD

02 Web App Deployment

03 Coverage

04 Mutation Testing

05 Refactoring

06 Benchmarking

07 Test Suite Generation

08 Vulnerabilities

Report

Source code