Apache Commons Imaging

https://github.com/apache/commons-imaging

Francesco Pagano - 0522501711

Introduction

Introduction

Apache Commons Imaging is a Java library designed for **working with image files** and is a part of the Apache Commons project.

The library provides tools to read and write various image formats, including JPEG, PNG, BMP, GIF, TIFF, and more.

The Commons Imaging project size is **31k LOC** (lines of code) with **448 classes**.

Software Quality Analysis

https://www.sonarsource.com/products/sonarcloud/

SonarCloud

139 bugs

- 62 critical
- 77 minor

(All Reliability)

2.2k code smells

(All Maintainability)

SonarCloud

9 critical bugs: *ArrayIndexOutOfBoundsException*

Fix:
To address these bugs, I used a conditional operation to make sure that the array was not empty before trying to get elements from it

52 critical bugs: *Remove this reference to "FieldTypeX"* (Not fixed)

SonarCloud

71 minor bugs: *Remove this useless shift*

Fix:
I considered these as false positives because even though shifting by 0 does not do anything it improves readability of the code when other shifting operations are performed on the same line of code.

6 minor bugs: *Cast operations*

Fix:
Adding the appropriate cast

SonarCloud

Medium code smells: *Remove this commented out code*

Fix:
Removed this code

SonarCloud

Minor code smells: *ImagingException subclass of IOException*

Fix:
Removing the superfluous exceptions

Minor code smells: *"public" modifier for test classes and methods*

Fix:
Removing the public modifier

SonarCloud

Solved

12 critical bugs
77 minor bugs
&
1.2k code smells

Docker

https://www.educative.io/answers/how-do-you-dockerize-a-maven-project

Docker

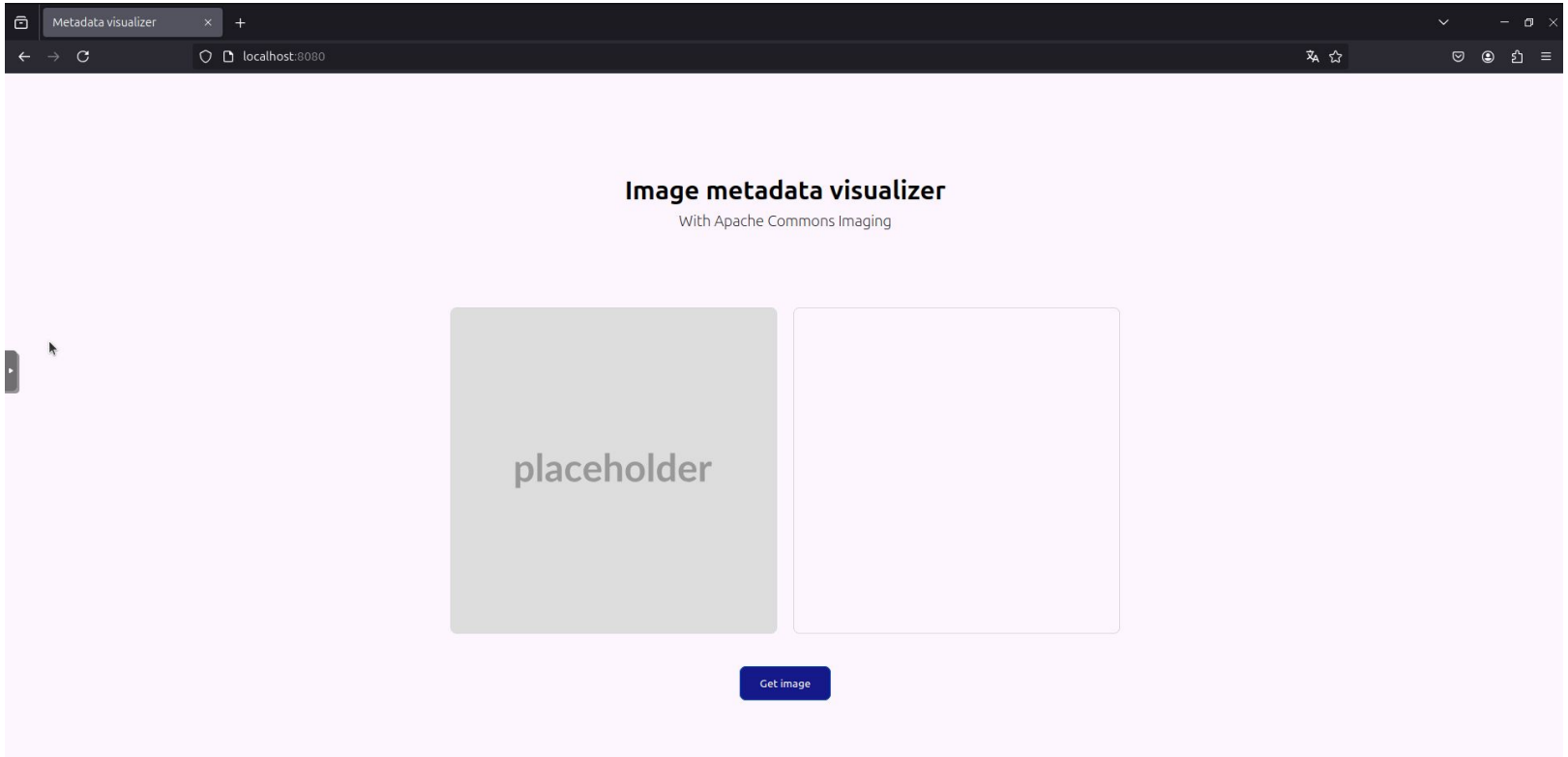
An image is taken from the website https://thispersondoesnotexist.com and saved in jpg format, initially without metadata.

Metadata relating to the position are added in a new image, such as latitude and longitude.

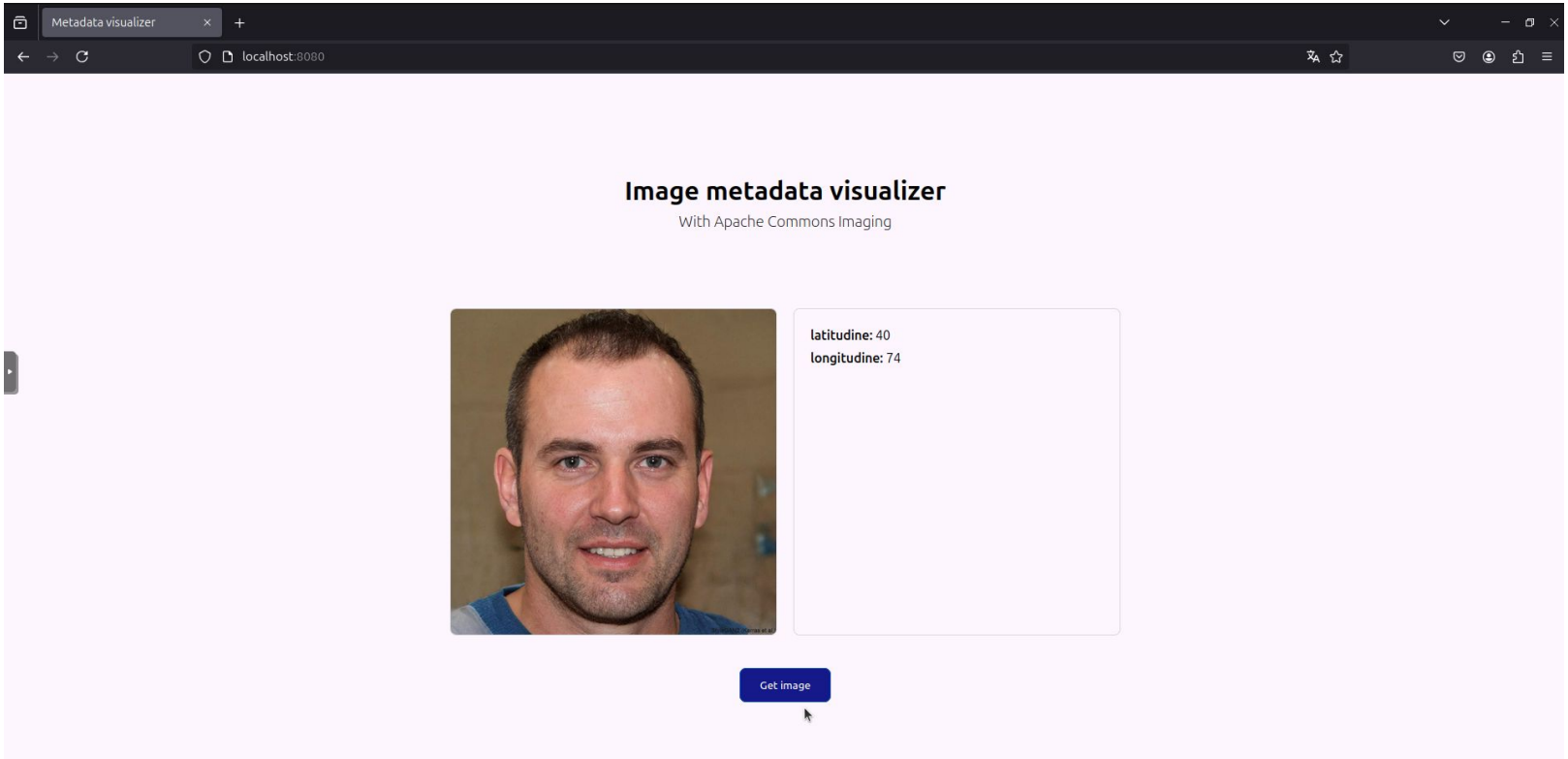
An HTML page exposed on port 8080, every time the get-image button is pressed the latest image taken from the website is shown with the related metadata.

This Docker image is available on: https://hub.docker.com/r/francescopagano45/example-docker.jar

Docker


Metadata visualizer
localhost:8080
Image metadata visualizer
With Apache Commons Imaging
placeholder
Get image

Docker


Metadata visualizer
localhost:8080
Image metadata visualizer
With Apache Commons Imaging
latitudine: 40
longitudine: 74
Get image

Code Coverage Analysis

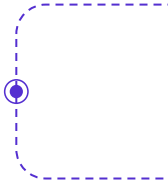
https://www.jacoco.org/jacoco/trunk/doc/maven.html

https://docs.codecov.com/docs/github-tutorial

Jacoco

44 Packages

Instructions coverage = 77%
branch coverage = 64%



Apache Commons Imaging

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
org.apache.commons.imaging.formats.tiff		57%		51%	385	789	618	1.766	70	296	1	29
org.apache.commons.imaging.formats.tiff.write		66%		63%	159	381	251	1.045	38	144	0	12
org.apache.commons.imaging.formats.png		75%		72%	92	301	183	895	14	116	1	21
org.apache.commons.imaging.common		80%		66%	166	462	235	1.035	26	197	0	20
org.apache.commons.imaging.palette		71%		65%	90	242	159	699	11	69	0	16
org.apache.commons.imaging		73%		60%	139	318	210	642	70	197	1	16
org.apache.commons.imaging.formats.pcx		66%		61%	82	167	118	421	9	38	0	6
org.apache.commons.imaging.formats.bmp		72%		61%	69	176	141	633	4	53	0	13
org.apache.commons.imaging.internal		28%		25%	63	80	107	169	22	35	0	3
org.apache.commons.imaging.formats.jpeg		74%		44%	176	300	203	580	15	73	0	12
org.apache.commons.imaging.formats.psd		68%		46%	56	95	125	308	13	41	1	6
org.apache.commons.imaging.formats.pnm		69%		59%	62	182	140	478	13	91	2	14
org.apache.commons.imaging.formats.gif		78%		75%	63	201	94	603	9	65	0	12
org.apache.commons.imaging.formats.icns		82%		61%	63	144	93	323	13	50	1	8
org.apache.commons.imaging.formats.xpm		74%		58%	70	127	104	381	4	31	0	4
org.apache.commons.imaging.formats.ico		72%		73%	38	98	73	346	12	38	1	9
org.apache.commons.imaging.formats.tiff.datareaders		89%		80%	75	237	66	701	3	30	0	5
org.apache.commons.imaging.color		89%		65%	75	208	103	691	7	100	0	11
org.apache.commons.imaging.formats.tiff.itu_t4		91%		82%	38	175	74	560	5	42	0	7
org.apache.commons.imaging.icc		83%		53%	39	84	65	277	16	53	0	11
org.apache.commons.imaging.formats.jpeg.segments		76%		46%	64	137	63	298	13	67	0	17
org.apache.commons.imaging.formats.tiff.taginfos		78%		50%	41	137	60	253	23	115	1	38
org.apache.commons.imaging.formats.jpeg.iptc		86%		64%	43	107	60	309	8	50	0	8
org.apache.commons.imaging.formats.tiff.fieldtypes		76%		67%	24	79	54	206	7	34	0	9
org.apache.commons.imaging.formats.psd.dataparsers		33%		66%	13	27	39	64	12	24	2	8
org.apache.commons.imaging.formats.tiff.photometricinterpreters		73%		100%	5	34	34	158	5	23	1	10
org.apache.commons.imaging.formats.png.chunks		80%		75%	25	79	28	186	14	57	0	12
org.apache.commons.imaging.formats.xbm		78%		60%	36	72	38	186	4	23	0	3
org.apache.commons.imaging.formats.jpeg.xmp		73%		61%	22	70	40	160	11	44	0	8
org.apache.commons.imaging.bytesource		74%		58%	27	64	31	123	2	24	0	4
org.apache.commons.imaging.formats.jpeg.decoder		96%		91%	19	155	19	609	1	40	0	6
org.apache.commons.imaging.formats.webp		89%		80%	17	72	12	167	0	29	0	6
org.apache.commons.imaging.mylzw		92%		85%	19	100	20	271	6	48	0	6
org.apache.commons.imaging.formats.wbmp		79%		79%	9	34	13	91	4	22	0	3
org.apache.commons.imaging.formats.psd.datareaders		61%		50%	5	10	18	43	2	4	1	2
org.apache.commons.imaging.formats.jpeg.exif		85%		86%	11	42	23	138	7	27	0	6
org.apache.commons.imaging.formats.rgbe		90%		78%	10	46	8	115	2	25	0	4
org.apache.commons.imaging.formats.webp.chunks		93%		69%	15	59	12	128	2	38	2	11
org.apache.commons.imaging.formats.dcx		78%		71%	8	25	11	56	4	18	0	2
org.apache.commons.imaging.formats.tiff.photometricinterpreters.floatingpoint		94%		86%	15	70	11	161	3	27	0	3
org.apache.commons.imaging.exampleDocker		0%		n/a	2	2	8	8	2	2	1	1
org.apache.commons.imaging.formats.tiff.constants		99%		28%	4	33	5	539	1	29	0	22
org.apache.commons.imaging.formats.png.scanlinefilters		100%		100%	0	26	0	58	0	11	0	5
org.apache.commons.imaging.formats.png.transparencyfilters		100%		90%	1	14	0	31	0	9	0	4
Total	21.454 of 96.632	77%	2.545 of 7.163	64%	2.435	6.261	3.769	16.911	507	2.549	16	433

Jacoco

Apache Commons Imaging > org.apache.commons.imaging

org.apache.commons.imaging

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
ColorTools		0%		0%	21	21	62	62	16	16	1	1
AbstractImageParser		59%		47%	33	56	41	86	19	37	0	1
Imaging		84%		71%	38	101	35	185	13	51	0	1
FormatCompliance		66%		75%	8	28	25	71	3	14	0	1
PixelDensity		25%		20%	21	27	21	33	11	17	0	1
ImageDump		62%		28%	9	17	11	35	0	7	0	1
ImagingParameters		48%		n/a	4	10	7	14	4	10	0	1
ImageInfo		98%		100%	1	25	2	71	1	23	0	1
ImagingFormatException		54%		n/a	1	2	2	4	1	2	0	1
ImagingRuntimeException		44%		n/a	1	2	2	4	1	2	0	1
ImagingException		97%		100%	1	15	2	29	1	5	0	1
ImageFormats		99%		50%	1	6	0	30	0	5	0	1
ImageInfo.CompressionAlgorithm		100%		n/a	0	3	0	9	0	3	0	1
ImageInfo.ColorType		100%		n/a	0	3	0	6	0	3	0	1
ImagingOverflowException		100%		n/a	0	1	0	2	0	1	0	1
ImagingConstants		100%		n/a	0	1	0	1	0	1	0	1
Total	1.025 of 3.841	73%	93 of 236	60%	139	318	210	642	70	197	1	16

Classes

Instructions coverage = 73%
branch coverage = 60%

Codecov

Branch Coverage = 71.48%
12096 of 16920 lines covered

bytesource	123	82	10	31	66.67%
color	691	535	53	103	77.42%
common	1035	718	81	236	69.37%
exampleDocker	8	0	0	8	0.00%
formats	13004	9374	798	2832	72.09%
icc	277	195	17	65	70.40%
internal	169	48	14	107	28.40%
mylzw	271	241	10	20	88.93%
palette	699	506	34	159	72.39%
AbstractImageParser.java	86	37	8	41	43.02%
ColorTools.java	62	0	0	62	0.00%
FormatCompliance.java	71	43	3	25	60.56%
ImageDump.java	35	21	3	11	60.00%
ImageFormats.java	30	29	1	0	96.67%
ImageInfo.java	86	84	0	2	97.67%
Imaging.java	186	134	17	35	72.04%
ImagingConstants.java	1	1	0	0	100.00%
ImagingException.java	29	27	0	2	93.10%
ImagingFormatException.java	4	2	0	2	50.00%
ImagingOverflowException.java	2	2	0	0	100.00%
ImagingParameters.java	14	7	0	7	50.00%
ImagingRuntimeException.java	4	2	0	2	50.00%
PixelDensity.java	33	8	4	21	24.24%

Mutation Testing

https://www.baeldung.com/java-mutation-testing-with-pitest

PITest

Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
11	84%	583/693	53%	432/819	63%	432/690

Breakdown by Package

Name	Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
org.apache.commons.imaging.color	11	84%	583/693	53%	432/819	63%	432/690

Report generated by PIT 1.15.3

Enhanced functionality available at arcmutate.com

Packages

PITest

Pit Test Coverage Report

Package Summary

org.apache.commons.imaging.color

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
11	84%	583/693	53%	432/819	63%	432/690

Breakdown by Class

Name	Line Coverage		Mutation Coverage		Test Strength	
ColorCieLab.java	83%	24/29	29%	7/24	37%	7/19
ColorCieLch.java	83%	24/29	29%	7/24	37%	7/19
ColorCieLuv.java	83%	24/29	29%	7/24	37%	7/19
ColorCmy.java	88%	28/32	38%	9/24	45%	9/20
ColorCmyk.java	84%	31/37	27%	8/30	33%	8/24
ColorConversions.java	88%	350/397	64%	364/573	71%	364/512
ColorDin99Lab.java	21%	5/24	0%	0/24	100%	0/0
ColorHsl.java	86%	25/29	38%	9/24	45%	9/20
ColorHsv.java	83%	24/29	29%	7/24	37%	7/19
ColorHunterLab.java	83%	24/29	29%	7/24	37%	7/19
ColorXyz.java	83%	24/29	29%	7/24	37%	7/19

Report generated by PIT 1.15.3

Classes

Automatic Test Case Generation

https://github.com/emaiannone/tools-tutorial/tree/master/randoop

Randoop

testclass=org.apache.commons.imaging.color.ColorConversions

1389 tests passed successfully

Performance Testing

https://www.baeldung.com/java-microbenchmark-harness

Java Microbenchmark Harness (JMH)

```
Result "benchmark.BenchmarkRunner.benchmarkLoadImage":
  136,447 ±(99.9%) 3,159 ms/op [Average]
  (min, avg, max) = (133,097, 136,447, 151,574), stdev = 4,217
  CI (99.9%): [133,288, 139,606] (assumes normal distribution)
```

```
Result "benchmark.BenchmarkRunner.benchmarkReadMetadata":
  1,815 ±(99.9%) 0,014 ms/op [Average]
  (min, avg, max) = (1,778, 1,815, 1,845), stdev = 0,018
  CI (99.9%): [1,801, 1,828] (assumes normal distribution)
```

Benchmark	Mode	Cnt	Score	Error	Units
BenchmarkRunner.benchmarkLoadImage	avgt	25	136,447	± 3,159	ms/op
BenchmarkRunner.benchmarkReadMetadata	avgt	25	1,815	± 0,014	ms/op

Software Vulnerabilities

https://github.com/emaiannone/tools-tutorial/tree/master/findsecbugs

https://github.com/emaiannone/tools-tutorial/tree/master/owaspdc

Find Security Bugs

Metrics

128129 lines of code analyzed, in 2762 classes, in 159 packages.

Metric	Total	Density*
High Priority Warnings	19	0.15
Medium Priority Warnings	95	0.74
Total Warnings	**114**	**0.89**

(Defects per Thousand lines of non-commenting source statements)*

Contents

Summary

Warning Type	Number
Security Warnings	114
Total	**114**

OWASP DC

How to read the report | Suppressing false positives | **Getting Help:** github issues

Sponsor

Project:

Scan Information (show all):

- *dependency-check version*: 8.2.1
- *Report Generated On*: Thu, 26 Dec 2024 12:53:11 +0100
- *Dependencies Scanned*: 14 (11 unique)
- *Vulnerable Dependencies*: 2
- *Vulnerabilities Found*: 2
- *Vulnerabilities Suppressed*: 0
- ...

Summary

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
example-docker.jar	cpe:2.3:a:apache:commons_imaging:1.0.0:alpha:*:*:*:*:*:* cpe:2.3:a:apache:commons_net:1.0.0:alpha:*:*:*:*:*:*		MEDIUM	1	Low	31
original-example-docker.jar	cpe:2.3:a:apache:commons_imaging:1.0.0:alpha:*:*:*:*:*:* cpe:2.3:a:apache:commons_net:1.0.0:alpha:*:*:*:*:*:*		MEDIUM	1	Low	32

Thank you For your attention!

https://github.com/Francesco-Pagano/commons-imaging