

# **Executive Report**

Project Name Portfolio

Size Rating



Lines of Code 275

Branch pmain

Version 0.0.1-SNAPSHOT

Analysis Date 2023-12-30

**Quality Gate** 

PASSED

#### Reliability



Bugs

0

Reliability on new code

New Bugs

Security



**Vulnerabilities** 

0

Security Hotspots

0

wity on new Lee Con

An Aniversia

**New Security Hotspot** 

Maintainability



16

Debt Ratio

0.6%

Maintainability on new code

New Code Smells

Debt Ratio on New Code

# Issues By Severing



Blocker
 Critical
 Major

CriticalMajorMinor

1 Info

Bug Sulne

Coverage



0%

**Unit Tests** 

1

Coverage on New Code

0

**Duplications** 



0%

**Duplicated blocks** 

0

Duplications on New Code







**Analysis Date** 

2023-12-30

Project Name Branch p main Portfolio Version 0.0.1-SNAPSHOT

Bugs	Vulnerabilities
0-	
	le resio.
	Full V
Code Smells	Security Hotspots
o report one	der purchasing
Duplications (Lines of code / Duplicated lines)	Duplicated Lines (%)
PIEBSE IF YOU IIKE	
Coverage (Lines to cover / Covered lines)	Coverage (%)
48	



## **Issues Breakdown**



Project Name Portfolio Branch Version p main 0.0.1-SNAPSHOT

Analysis Date 2023-12-30

## **Top Common Issues**

Severity 0 5 Field names should comply with a naming convention (java) Avoid field dependency injection (java) 0 Unnecessary imports should be removed (java) Local variables should not be declared and then immediately returned or thrown (java Unused "private" methods should be removed (java) 1 Public constants and fields initialized at declaration should be "static final" rather 1 Sections of code should not be commented out (java) 1 Tests should include assertions (java)

age. I

**Severity Levels** 

BlockerCriticalMajor

Minor

1 Info

Blocker Multiple severities found

**Issue Types** 

₩ Bug

← Vulnerability

① Security Hotspot





## **Issues Breakdown**



Project Name Portfolio Branch Version p main 0.0.1-SNAPSHOT

Analysis Date 2023-12-30

# **Top Issues By Severity**

Tests should include assertions (java)

Avoid field dependency injection (java)

Unused "private" methods should be removed (java)

Sections of code should not be commented out (java)

Field names should comply with a naming convention (java)

Unnecessary imports should be removed (java)

Local variables should not be declared and then immediately returned or thrown (nva)

Public constants and fields initialized at declaration should be "shatic final" rather than ... (java)

, 285e, 1

**Severity Levels** 

BlockerCriticalMajor

Minor

1 Info

Blocker Multiple severities found

**Issue Types** 

₩ Bug

€ Vulnerability

① Security Hotspot





# Security Hotspots Breakdown



**Project Name** Portfolio

Branch Version p main 0.0.1-SNAPSHOT **Analysis Date** 2023-12-30

# **Top Security Hotspots to Review**

Severity Rule

No Security Hoteno

**Severity Levels** 

Critical

Major

Minor

1 Info

Blocker Multiple severities found

**Issue Types** 

₩ Bug

← Vulnerability

Security Hotspot

bite **garden** 



# Security Hotspots Breakdown



Project Name Portfolio Branch Version main 0.0.1-SNAPSHOT

Analysis Date 2023-12-30

## **Top Security Hotspots Reviewed**

**O** acknowledged

 $\mathbf{0}$  fixed

0 safe

O of 0 reviewed





# reviewed

# **Top reviewed hotspots**

Severity

Rule

No Security Reviewed Hotspots found

Lease,

**Severity Levels** 

BlockerCriticalMajor

MinorInfo

Blocker Multiple severities found

**Hotspot Status** 

Acknowledged

SafeFixed







**Project Name** Portfolio

Branch Version 💪 main 0.0.1-SNAPSHOT **Analysis Date** 2023-12-30

### **Tests should include assertions**

Language java



Issues

Why is this an issue? A test case without assertions ensures only that no exceptions are throw?

junit tests

**Issues Breakdown** 

src/test/java/com/portfolioapp/PortfolioApplicationTests.java Add at least one assertion to this test case.





Project NameBranchp mainAnalysis DatePortfolioVersion0.0.1-SNAPSHOT2023-12-30

Rule

### **Avoid field dependency injection**

Language java **₩** 

Issues

Why is this an issue? Dependency injection frameworks such as Spring support dependency injection using annotations such as @Inject and @Autowired.

#### **Issues Breakdown**

- src/main/java/com/portfolioapp/controller/PortfolioController.java Remove this field injection and use constructor injection instead.
- src/main/java/com/portfolioapp/service/PortfolioService.java

  Remove this field injection and use constructor injection instead
- src/main/java/com/portfolioapp/service/Portfolioservice ava Remove this field injection and use constructor injection inste

L18

L17

L19

Lease,





Project Name Portfolio Branch Version p main

0.0.1-SNAPSHOT

Analysis Date 2023-12-30



Rule

### Unused "private" methods should be removed

Language java



Issues

Why is this an issue? private methods that are never executed are dead code: unnecessary, inoperative code that should be removed.



unused

#### **Issues Breakdown**

src/main/java/com/portfolioapp/service/PortfolioService.java

Remove this unused private "mapToPortfolioResponse" method.

L64

25e, 111





**Project Name** Portfolio

Branch Version 💪 main 0.0.1-SNAPSHOT **Analysis Date** 2023-12-30

### Sections of code should not be commented out

Language java



Issues

Why is this an issue? Commented-out code distracts the focus from the actual executed code

unused

**Issues Breakdown** 

src/main/java/com/portfolioapp/service/PortfolioService.java This block of commented-out lines of code should be removed.





**Project Name Analysis Date** Branch nain 🎖 Portfolio 2023-12-30 Version 0.0.1-SNAPSHOT



### Field names should comply with a naming convention

Language java



Issues

Why is this an issue? A naming convention in software development is a set of guidelines for partial of the set of guidelines for guidelines for partial of the set of guidelines for guide elements like variables, functions, and classes.



convention

#### **Issues Breakdown**

src/main/java/com/portfolioapp/entities/Portfolio.java Rename this field "Experience" to match the regular expression '^[a-z][a-zA-XC L22

src/main/java/com/portfolioapp/entities/Portfolio.java Rename this field "Project" to match the regular expression '^[a 2 L23

src/main/java/com/portfolioapp/entities/PortfolioResponse.jav Rename this field "Experience" to match the regular expression

L15

src/main/java/com/portfolioapp/entities/PortfolioResponse.java

L16

Rename this field "Project" to match the regular expression '^[a-z][a-zA-\tilde{z}-9]\*\$'

src/main/java/com/portfolioapp/service/portfolioService.java Please if you like it Rename this field "URL" to match the regular expression \( \) -z\[ \] zA-Z0-9]\*\$'.





**Project Name Analysis Date** Branch 🎾 main Portfolio 2023-12-30 Version 0.0.1-SNAPSHOT

### Unnecessary imports should be removed

Language java

**Issues** 3

Why is this an issue? Unnecessary imports refer to importing types that are not used or referenced anywith

unused

#### **Issues Breakdown**

- src/main/java/com/portfolioapp/controller/PortfolioController.java Remove this unused import 'org.springframework.http.HttpStatus'.
- src/main/java/com/portfolioapp/controller/PortfolioController/eva Remove this unused import 'org.springframework.http.ResponseEntity'.
- src/main/java/com/portfolioapp/repository/PortfolioRepository Remove this unused import 'org.springframework data.jpa.rep

L7

L8



Code Smell S1488



Project Name Portfolio Branch Version p main 0.0.1-SNAPSHOT

Analysis Date 2023-12-30



Rule

# Local variables should not be declared and then immediately returned or thrown

Language java **₩** 

Issues

Why is this an issue? Declaring a variable only to immediately return or throw it is considered a practice because it adds unnecessary complexity to the code.



clumsy

**Issues Breakdown** 

src/main/java/com/portfolioapp/controller/PortfolioController.java
Immediately return this expression instead of assigning it to the temporary variable "addedPortfolio".

L33

ase.



S1170

Project NameBranchp mainAnalysis DatePortfolioVersion0.0.1-SNAPSHOT2023-12-30

•

Rule

# Public constants and fields initialized at declaration should be "static final" rather than merely "final"

Language java 98

Issues

Why is this an issue? Making a public constant just final as opposed to static final leads to durlicating its value for every instance of the class, uselessly increasing the amount of memory required to execute the application.

convention

**Issues Breakdown** 

src/main/java/com/portfolioapp/service/PortfolioService.java Make this final field static too. L21

Jease,