

Java Forum Stuttgart, 13.07.2023

Raus aus der Wartungshölle ... zumindest ein bißchen

Sandra Parsick

@SandraParsick

@sparsick@mastodon.social

mail@sandra-parsick.de

Wer bin ich?

- Sandra Parsick
- Freiberuflicher Softwareentwickler und Consultant im Java-Umfeld
- Schwerpunkte:
 - Java Enterprise Anwendungen
 - Agile Methoden
 - Software Craftmanship
 - Automatisierung von Entwicklungsprozessen
- Trainings
- Workshops

 mail@sandra-parsick.de

 @SandraParsick

 @sparsick@mastodon.social

 <https://www.sandra-parsick.de>

 <https://ready-for-review.dev>





Wieso haben wir
das so gemacht?

Welche
Abhängigkeiten
haben wir im
Einsatz?

Wie können wir
am besten up to
date bleiben?

Entspricht die
Implementierung
noch den
Architekturbildern?

Welche
Softwarelizenzen
haben wir im
Einsatz?

Sind wir von
Sicherheitslücken
betroffen?

Onboarding neuer
Devs

Wie bekommen
wir die
Anwendung lokal
zum Laufen?

Willkommen in der Wartungshölle



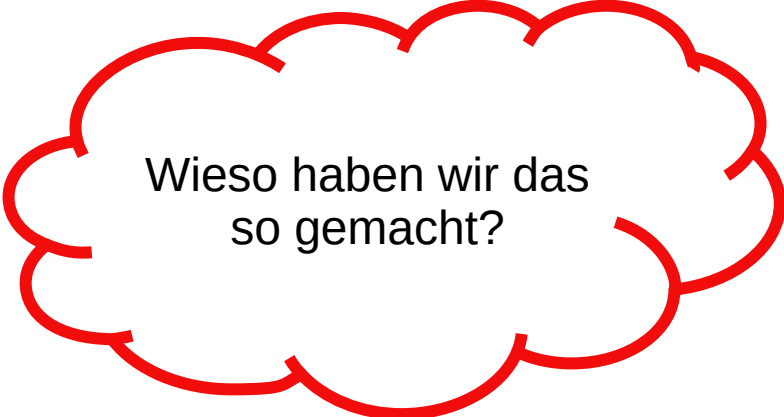








Automation is Queen :)



Wieso haben wir das
so gemacht?

Architektur dokumentieren

1. Einführung und Ziele

- 1.1 Aufgabenstellung
- 1.2 Qualitätsziele
- 1.3 Stakeholder

2. Randbedingungen

- 2.1 Technische Randbedingungen
- 2.2 Organisatorische Randbedingungen
- 2.3 Konventionen

3. Kontextabgrenzung

- 3.1 Fachlicher Kontext
- 3.2 Technischer- oder Verteilungskontext

4. Lösungsstrategie

5. Bausteinsicht

- 5.1 Ebene 1
- 5.2 Ebene 2

....

6. Laufzeitsicht

- 6.1 Laufzeitszenario 1
- 6.2 Laufzeitszenario 2

....

7. Verteilungssicht

- 7.1 Infrastruktur Ebene 1
- 7.2 Infrastruktur Ebene 2

....

8. Querschnittliche Konzepte

- 8.1 Fachliche Struktur und Modelle
- 8.2 Architektur- und Entwurfsmuster
- 8.3 Unter-der-Haube
- 8.4 User Experience

....

9. Entwurfsentscheidungen

- 9.1 Entwurfsentscheidung 1
- 9.2 Entwurfsentscheidung 2

....

10. Qualitätsanforderungen

- 10.1 Qualitätsbaum
- 10.2 Qualitätsszenarien

11. Risiken und technische Schulden

12. Glossar



Was ist wenn nix da ist?

Source Code ist immer da.



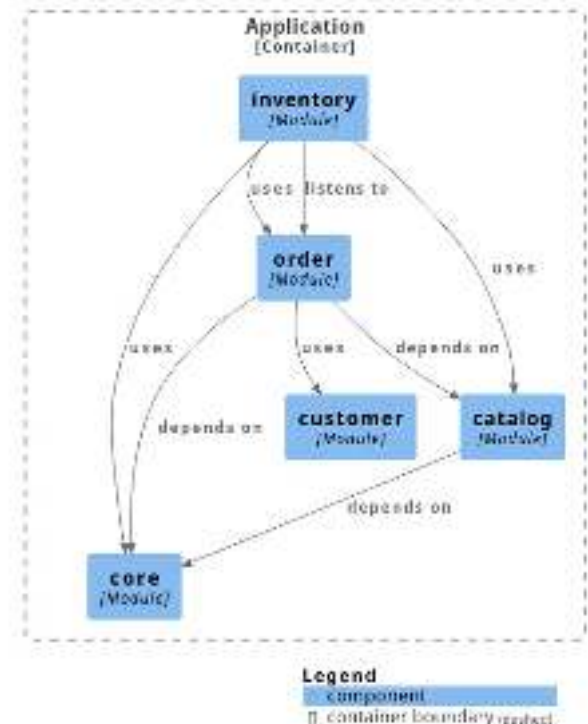
Spring Modulith

@springmodulith

Modulithic java applications using Spring Boot

- Spring Projekt
- Basiert auf jMolecule
- Spring Boot Erweiterung
- einer der Features:
u.a. Generierung von Diagrammen
basierend auf Annotationen

```
class DocumentationTests {  
  
    ApplicationModules modules = ApplicationModules.of(Application.class);  
  
    @Test  
    void writeDocumentationSnippets() {  
  
        new Documenter(modules)  
            .writeModulesAsPlantUml()  
            .writeIndividualModulesAsPlantUml();  
    }  
}
```



Entscheidungen dokumentieren

ADR (architecture decision records)

<https://adr.github.io/>

[short title of solved problem and solution]

- Status: [proposed | rejected | accepted | deprecated | ... | superseded by [ADR-0005](0005-example.adoc)] <!-- optional -->
- Decision makers: [list everyone involved in the decision] <!-- optional -->
- Date: [YYYY-MM-DD when the decision was last updated] <!-- optional -->

Technical Story: [description | ticket/issue URL] <!-- optional -->

Context and Problem Description

Decision driver <!-- optional -->

- [driver 1, e.g., a force, facing concern, ...]
- [driver 2, e.g., a force, facing concern, ...]
- ... <!-- numbers of drivers can vary -->

Considered Options

- [option 1]
- [option 2]
- [option 3]
- ... <!-- numbers of options can vary -->

Decision

Chosen option: "[option 1]", because [justification. e.g., only option, which meets k.o. criterion decision driver | which resolves force force | ... | comes out best (see below)].

Positive Consequences

- [e.g., improvement of quality attribute satisfaction, follow-up decisions required, ...]
- ...

Negative Consequences

- [e.g., compromising quality attribute, follow-up decisions required, ...]
- ...

Pros and Cons of the Options

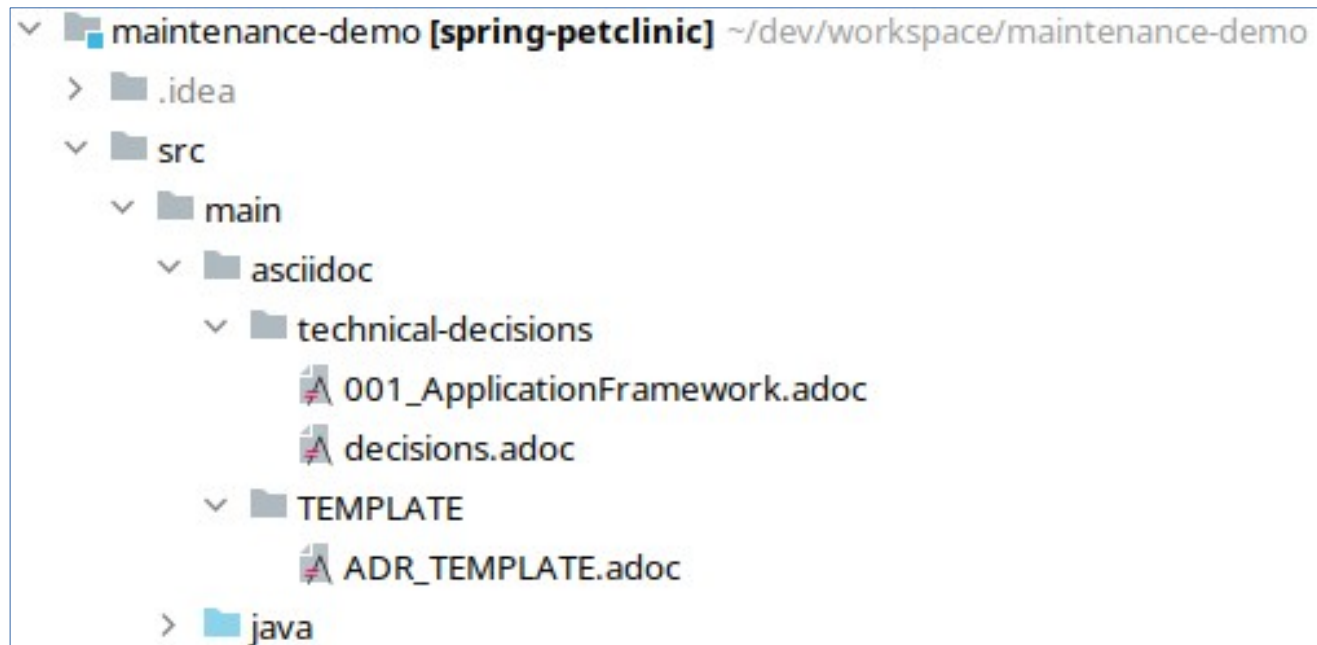
[option 1]

[example | description | pointer to more information | ...] <!-- optional -->

- **Good:** because [argument a]
- **Good:** because [argument b]
- **Bad:** because [argument c]
- ... <!-- numbers of pros and cons can vary -->

Links <!-- optional -->

- [Link type] [Link to ADR] <!-- example: Refined by [ADR-0005](0005-example.adoc) -->
- ... <!-- numbers of links can vary -->

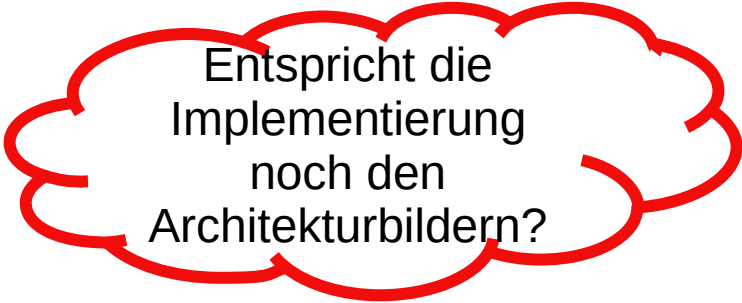




```
<plugin>  
  <groupId>org.asciidoctor</groupId>  
  <artifactId>asciidoctor-maven-plugin</artifactId>  
</plugin>
```

Viele ADR Templates im Netz

- ADR template by Michael Nygard (simple and popular)
- ADR template by Jeff Tyree and Art Akerman (more sophisticated)
- ADR template for Alexandrian pattern (simple with context specifics)
- ADR template for business case (more MBA-oriented, with costs, SWOT, and more opinions)
- ADR template MADR (more Markdown)
- ADR template using Planguage (more quality assurance oriented)



Entspricht die
Implementierung
noch den
Architekturbildern?

Spring Pet Clinic

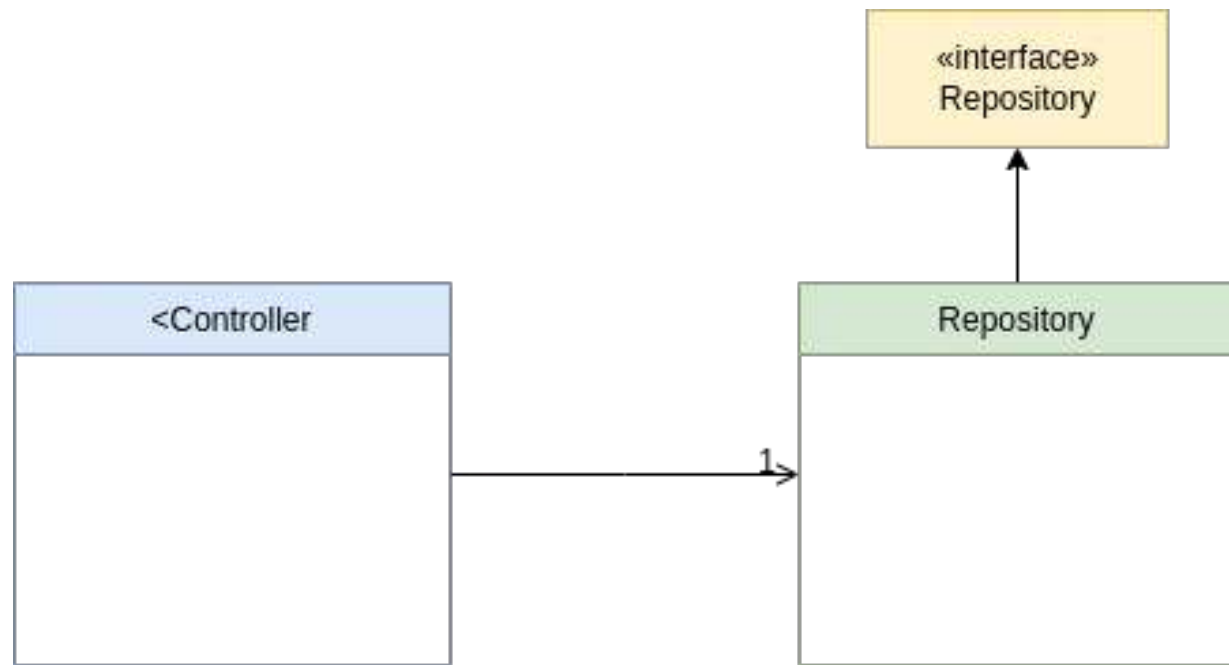
[HOME](#)[FIND OWNERS](#)[VETERINARIANS](#)[ERROR](#)

Owners

Name	Address	City	Telephone	Pets
Jeff Black	1450 Oak Blvd.	Monona	608555387	Lucky
Jean Coleman	105 N. Lake St.	Monona	6085552654	Max Samantha
Betty Davis	638 Cardinal Ave.	Sun Prairie	6085551749	Basil
Harold Davis	563 Friendly St.	Windsor	6085553196	IggY
Maria Escobedo	345 Maple St.	Madison	6085557683	Mulligan
Carlos Estraban	2335 Independence La.	Waunakee	6085555487	Lucky Siy
George Franklin	110 W. Liberty St.	Madison	6085551023	Leo
Peter McFavish	2387 S. Fair Way	Madison	6085552765	George
Eduardo Rodriguez	2693 Commerce St.	McFarland	6085558763	Jewel Rosy
David Schroeder	2749 Blackhawk Trail	Madison	6085559435	Freddy

Spring Pet Clinic

- Technologiestack:
 - Java 8
 - Spring Boot 2.2.0
 - JUnit 4
 - Maven





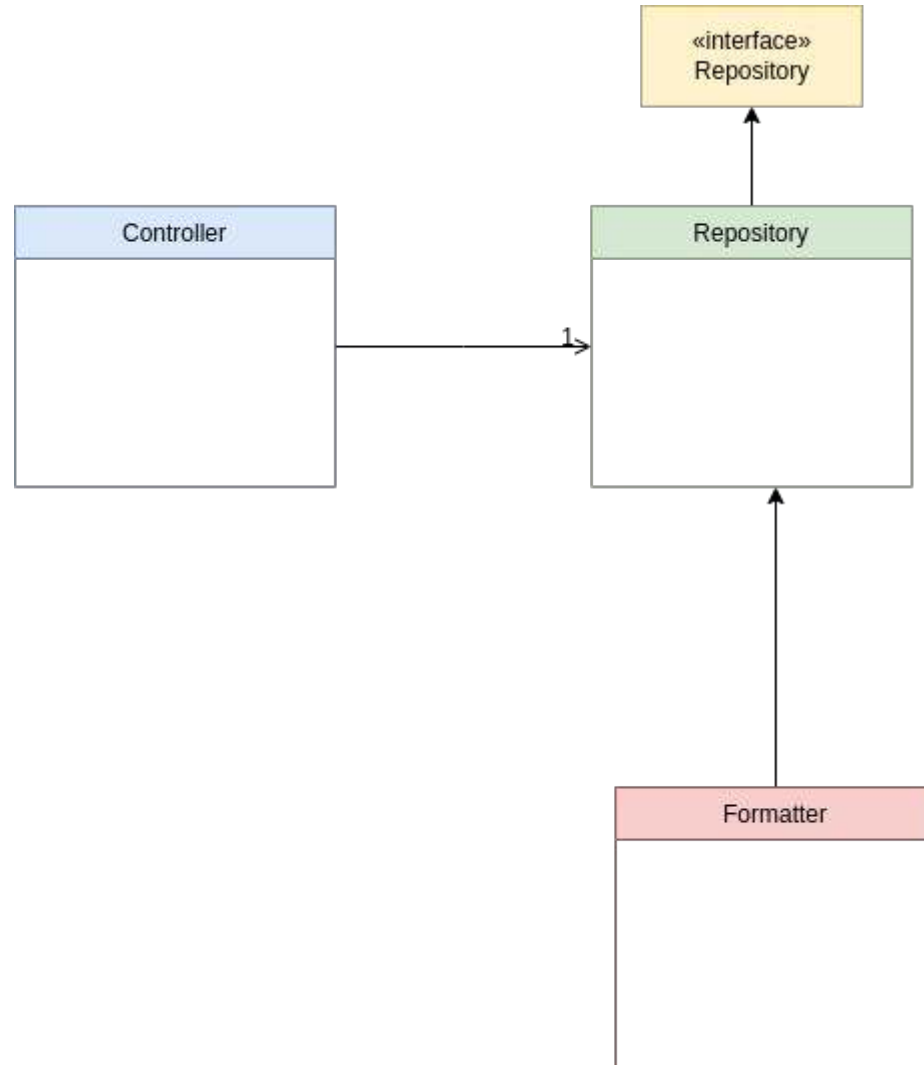


```
@AnalyzeClasses(packages =
{"org.springframework.samples.petclinic.owner", "org.springframework.samples.petclinic.vet"},
importOptions = ImportOption.DoNotIncludeTests.class)
public class ControllerRepositoryArchTest {

    @ArchTest
    ArchRule relation = classes()
        .that().haveNameMatching(".*Controller")
        .should().accessClassesThat().haveNameMatching(".*Repository");

    @ArchTest
    ArchRule controllerAnnotation = classes()
        .that().haveNameMatching(".*Controller")
        .should().beAnnotatedWith("org.springframework.stereotype.Controller");

    @ArchTest
    ArchRule repositoryDefinition = classes()
        .that().haveNameMatching(".*Repository")
        .should().beAnnotatedWith("org.springframework.stereotype.Repository")
        .orShould().beAssignableTo("org.springframework.data.repository.Repository");
}
```





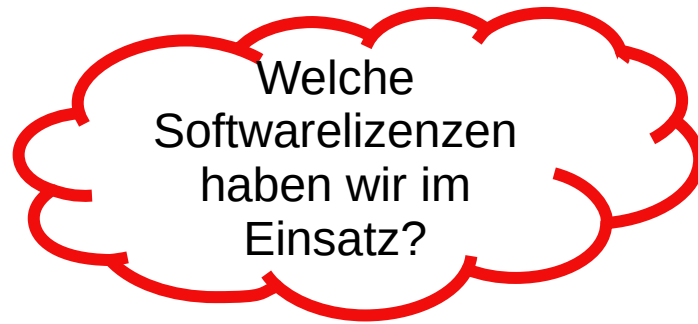
```
@AnalyzeClasses(importOptions = ImportOption.DoNotIncludeTests.class)
public class ControllerRepositoryFreezeArchTest {

    @ArchTest
    ArchRule relation = FreezingArchRule.freeze(classes()
        .that().haveNameMatching(".*Repository")
        .should().onlyBeAccessed().byClassesThat().haveNameMatching(".*Controller"));
}
```

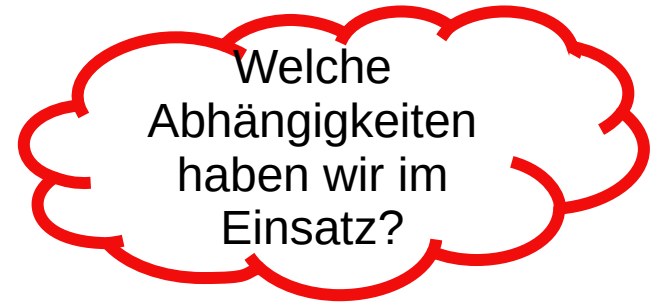



```
// archunit.properties
```

```
freeze.store.default.allowStoreCreation = true  
archRule.failOnEmptyShould = false
```



Welche
Softwarelizenzen
haben wir im
Einsatz?



Welche
Abhängigkeiten
haben wir im
Einsatz?

„Schnelle Küche“

-

mit Maven Boardmitteln

```
mvn dependency:analyze-report
```


Abhängigkeitsanalyse

Verwendete und deklarierte Abhängigkeiten

GroupId	ArtifactId	Version	Scope	Classifier	Type	Optional
javax.cache	cache-api	1.1.1	compile		jar	false
com.tngtech.archunit	archunit	1.0.1	test		jar	false

Verwendete aber undeklarierte Abhängigkeiten

GroupId	ArtifactId	Vers
org.springframework	spring-test	5.2.0
org.springframework	spring-core	5.2.0
org.springframework.boot	spring-boot	2.2.0
org.springframework	spring-webmvc	5.2.0
org.springframework.data	spring-data-commons	2.2.0
org.springframework	spring-context	5.2.0
org.springframework.boot	spring-boot-test-autoconfigure	2.2.0
org.springframework	spring-web	5.2.0
jakarta.validation	jakarta.validation-api	2.0.1
org.hamcrest	hamcrest	2.1
org.assertj	assertj-core	3.13.
org.springframework.boot	spring-boot-autoconfigure	2.2.0
org.springframework.data	spring-data-jpa	2.2.0
junit	junit	4.12
org.springframework.boot	spring-boot-test	2.2.0
org.mockito	mockito-core	3.1.0
jakarta.persistence	jakarta.persistence-api	2.2.3
org.springframework	spring-beans	5.2.0
com.tngtech.archunit	archunit-junit5-api	1.0.1
jakarta.xml.bind	jakarta.xml.bind-api	2.3.2
org.springframework	spring-tx	5.2.0
org.springframework	spring-orm	5.2.0

Unverwendete aber deklarierte Abhängigkeiten

GroupId	ArtifactId	Version	Scope	Classifier	Type	Optional
org.springframework.boot	spring-boot-starter-actuator	2.2.0.RELEASE	compile		jar	false
org.springframework.boot	spring-boot-starter-cache	2.2.0.RELEASE	compile		jar	false
org.springframework.boot	spring-boot-starter-data-jpa	2.2.0.RELEASE	compile		jar	false
org.springframework.boot	spring-boot-starter-web	2.2.0.RELEASE	compile		jar	false
org.springframework.boot	spring-boot-starter-thymeleaf	2.2.0.RELEASE	compile		jar	false
org.springframework.boot	spring-boot-starter-test	2.2.0.RELEASE	test		jar	false
org.hsqldb	hsqldb	2.5.0	runtime		jar	false
mysql	mysql-connector-java	8.0.18	runtime		jar	false
org.ehcache	ehcache	3.8.1	compile		jar	false
org.webjars	webjars-locator-core	0.41	compile		jar	false
org.webjars	jquery	2.2.4	compile		jar	false
org.webjars	jquery-ui	1.11.4	compile		jar	false
org.webjars	bootstrap	3.3.6	compile		jar	false
org.springframework.boot	spring-boot-devtools	2.2.0.RELEASE	compile		jar	
com.tngtech.archunit	archunit-junit5	1.0.1	test		jar	false

```
mvn license:license-list
```

[INFO] Available licenses :

- * agpl_v3 : GNU Affero General Public License (AGPL) version 3.0
- * apache_v2 : Apache License version 2.0
- * bsd_2 : BSD 2-Clause License
- * bsd_3 : BSD 3-Clause License
- * cddl_v1 : COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0
- * epl_only_v1 : Eclipse Public License - v 1.0
- * epl_only_v2 : Eclipse Public License - v 2.0
- * epl_v1 : Eclipse Public + Distribution License - v 1.0
- * epl_v2 : Eclipse Public License - v 2.0 with Secondary License
- * eupl_v1_1 : European Union Public License v1.1
- * fdl_v1_3 : GNU Free Documentation License (FDL) version 1.3
- * gpl_v1 : GNU General Public License (GPL) version 1.0
- * gpl_v2 : GNU General Public License (GPL) version 2.0
- * gpl_v3 : GNU General Public License (GPL) version 3.0
- * lgpl_v2_1 : GNU General Lesser Public License (LGPL) version 2.1
- * lgpl_v3 : GNU General Lesser Public License (LGPL) version 3.0
- * mit : MIT-License

[INFO] -----

[INFO] BUILD SUCCESS

[INFO] -----

[INFO] Total time: 6.328 s

[INFO] Finished at: 2022-12-12T19:42:48+01:00

[INFO] -----

```
mvn license:third-party-report
```

Overview

This report summarizes all third-party dependencies of the project.

# of total dependencies	121
# of dependencies with license declared in their pom	121
# of dependencies with no license in their pom, but filled in the third-party missing file	0
# of dependencies with no license at all	0

Third-parties list

Status	GroupId:ArtifactId:Version	Scope	Classifier	Type	License(s)
	antlr:antlr:2.7.7	compile		jar	BSD License
	ch.qos.logback:logback-classic:1.2.3	compile		jar	Eclipse Public License - v 1.0 GNU Lesser General Public License
	ch.qos.logback:logback-core:1.2.3	compile		jar	Eclipse Public License - v 1.0 GNU Lesser General Public License
	com.fasterxml.classmate:1.5.0	compile		bundle	The Apache Software License, Version 2.0
	com.fasterxml.jackson.core:jackson-annotations:2.10.0	compile		bundle	The Apache Software License, Version 2.0
	com.fasterxml.jackson.core:jackson-core:2.10.0	compile		bundle	The Apache Software License, Version 2.0
	com.fasterxml.jackson.core:jackson-databind:2.10.0	compile		bundle	The Apache Software License, Version 2.0
	com.fasterxml.jackson.datatype:jackson-datatype-jdk8:2.10.0	compile		bundle	The Apache Software License, Version 2.0
	com.fasterxml.jackson.datatype:jackson-datatype-jsr310:2.10.0	compile		bundle	The Apache Software License, Version 2.0
	com.fasterxml.jackson.module:jackson-module-parameter-names:2.10.0	compile		bundle	The Apache Software License, Version 2.0
	com.jayway.jsonpath:json-path:2.4.0	test		jar	The Apache Software License, Version 2.0
	com.sun.istack:istack-commons-runtime:3.0.8	compile		jar	Eclipse Distribution License - v 1.0
	com.sun.xml.fastinfoset:FastInfoset:1.2.16	compile		jar	Apache License, Version 2.0 Eclipse Distribution License - v 1.0
	com.tngtech.archunit:archunit:1.0.1	test		jar	BSD The Apache Software License, Version 2.0
	com.tngtech.archunit:archunit-junit5:1.0.1	test		jar	The Apache Software License, Version 2.0
	com.tngtech.archunit:archunit-junit5-api:1.0.1	test		jar	The Apache Software License, Version 2.0
	com.tngtech.archunit:archunit-junit5-engine:1.0.1	test		jar	The Apache Software License, Version 2.0
	com.tngtech.archunit:archunit-junit5-engine-api:1.0.1	test		jar	The Apache Software License, Version 2.0
	com.vaadin.external.google:android-json:0.0.20131108.vaadin1	test		jar	Apache License 2.0
	com.zaxxer:HikariCP:3.4.1	compile		bundle	The Apache Software License, Version 2.0

Neuer Silver Bullet?

-

SBOM



High-Level Object Model

Metadata	Supplier	Authors	Component	
	Manufacturer	Tools		
Components	Application	Framework	Operating System	Firmware
	Library	Container	Device	File
Services	Provider	Data		
	Endpoints	Trust Boundary		
Dependencies	Components	Services		
Compositions	Completeness of:			
	Components	Services	Dependencies	
Vulnerabilities	Details	Source	Exploitability	
	Advisories	Risk Ratings		
Extensions	Properties	Per Organization	Per Team	
	Formal Taxonomy	Per Industry	...	

License Compliance



```
<?xml version='1.0' encoding='UTF-8'?>
<bor xmlns='http://cyclonedx.org/schemas/bom/1.4'
  serialNumber='urn:uuid:3e571667-395a-41f5-a30f-a58321a69b73'
  version='1.1'>
  <components>
    <component type='library'>
      <group>org.acme</group>
      <name>card-verifier</name>
      <version>1.8.2</version>
      <licenses>
        <!-- This component has an SPDX license expression -->
        <expression>LPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0</expression>
      </licenses>
    </component>
    <component type='library'>
      <group>com.example</group>
      <name>util</name>
      <version>2.X.9</version>
      <licenses>
        <!-- This component has a license name with optional text -->
        <name>Example, Inc. Commercial License</name>
        <text content-type='text/plain'
          encoding='ascii'>vGAlTHRrRHlg7r0yTHRoZSRFeSFrrGx11CR1bn9uTFNvdwTlrm8pYAwgTGIj2W5z75Bab2vzTf01c
          nU=</text>
        </licenses>
      </component>
    </components>
  </bor>
```

Dependency Graph

```
<?xml version='1.0' encoding='UTF-8'?>
<bam xmlns='http://cyclonedx.org/schema/bom/1.4'
  serialNumber='urn:uuid:3e671687-395b-41f5-b30f-b58921a65679'
  version='1'>
  <metadata>
    <!-- The component for which this BOM describes -->
    <component type='application' bom-ref='acme-app'>
      <name>Acme Application</name>
      <version>9.1.1</version>
    </component>
  </metadata>
  <components>
    <component type='framework' bom-ref='pkg:maven/org.acme/web-framework@1.0.0'>
      <group>org.acme</group>
      <name>web-framework</name>
      <version>1.0.0</version>
      <url>pkg:maven/org.acme/web-framework@1.0.0</url>
    </component>
    <component type='library' bom-ref='pkg:maven/org.acme/persistence@3.1.0'>
      <group>org.acme</group>
      <name>persistence</name>
      <version>3.1.0</version>
      <url>pkg:maven/org.acme/persistence@3.1.0</url>
    </component>
    <component type='library' bom-ref='pkg:maven/org.acme/common-util@3.0.0'>
      <group>org.acme</group>
      <name>common-util</name>
      <version>3.0.0</version>
      <url>pkg:maven/org.acme/common-util@3.0.0</url>
    </component>
  </components>
  <dependencies>
```

Dependency Graph

```
<!-- component -->
<dependencies>
  <!-- Direct dependencies of the data application -->
  <dependency ref='acme-app'>
    <dependency ref='pkg:maven/org.acme/web-framework@1.0.0' />
    <dependency ref='pkg:maven/org.acme/persistence@3.1.0' />
  </dependency>
  <!-- All other dependency relationships -->
  <dependency ref='pkg:maven/org.acme/web-framework@1.0.0'>
    <dependency ref='pkg:maven/org.acme/common-util@3.0.0' />
  </dependency>
  <dependency ref='pkg:maven/org.acme/persistence@3.1.0'>
    <dependency ref='pkg:maven/org.acme/common-util@3.0.0' />
  </dependency>
  <dependency ref='pkg:maven/org.acme/common-util@3.0.0' />
</dependencies>
</component>
```

Tool Center

Open source tools that support CycloneDX

All tools 33

Open Source 124

Proprietary 13

Build Integration 12

Analysis 14

Author 2

GitHub Action 11

Transform 1

Library 12

Signing / Notary 12

Distribute 1

Open Source Build Integration

Auditjs

Webhook

Analyze an NPM package from file to identify known vulnerabilities

🔗 Fork 14 📄 Stars 128

Open Source Analysis

BOM Repository Server

🔗 Fork 1

A lightweight repository server used to publish, manage, and distribute cycloneDX SBOMs

🔗 Fork 6 📄 Stars 45

Open Source Analysis

Checkov

🔗 Fork 1

Prevent cloud misconfigurations during build-time for Terraform, CloudFormation, Kubernetes, Serverless Framework and other infrastructure as code languages with Checkov by Enghagen. Can be run in CycloneDX.

🔗 Fork 703 📄 Stars 5K

Open Source Build Integration

Chelsea

🔗 Fork 1

Dependency vulnerability auditor for Ruby

🔗 Fork 7 📄 Stars 10

Open Source Signing / Notary

Codenotary CAS

Codenotary

CAS is an open source attestation service for the community. Notarize and authorize files, directories, git repos and build SBOMs of containers. CAS natively supports CycloneDX SBOMs.

🔗 Fork 32 📄 Stars 186

Open Source Signing / Notary GitHub Action

Codenotary CAS Authenticate Docker Image and SBOM

Codenotary

A GitHub Action which authenticates notarized Docker images and SBOMs.

🔗 Fork 2 📄 Stars 2

Open Source Signing / Notary GitHub Action

Codenotary CAS Notarize Docker Image and SBOM

Codenotary

A GitHub Action which notarizes and creates an SBOM for Docker Images

🔗 Fork 1 📄 Stars 1

Open Source Signing / Notary

Codenotary vcn

Codenotary

Protects an organization's software development pipeline from supply chain attacks. Codenotary natively supports CycloneDX SBOMs.

🔗 Fork 32 📄 Stars 186

Open Source Signing / Notary

Cosign

Signum

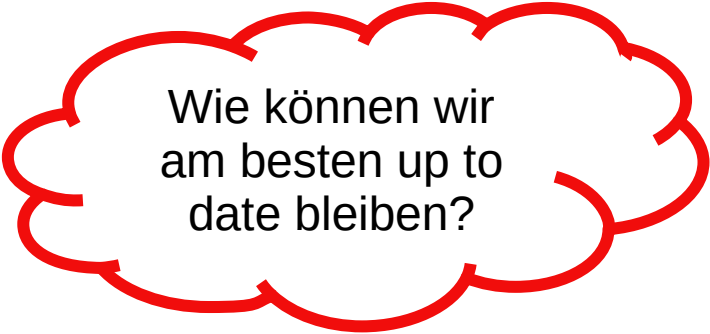
Container Signing, Verification and Storage in an OCI registry, including CycloneDX SBOMs

🔗 Fork 254 📄 Stars 24K


```
mvn cyclonedx:makeAggregateBom
```



```
[INFO] Scanning for projects...
[INFO]
[INFO] < org.springframework.samples:spring-petclinic >
[INFO] Building petclinic 2.2.8.BUILD-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- cyclonedx-naven-plugin:2.7.3:makeAggregateBom (default-cli) @ spring-petclinic ---
[INFO] CycloneDX: Parameters
[INFO] -----
[INFO] schemaVersion      : 1.4
[INFO] includeBomSerialNumber : true
[INFO] includeCompileScope  : true
[INFO] includeProvidedScope  : true
[INFO] includeRuntimeScope   : true
[INFO] includeTestScope     : false
[INFO] includeSystemScope    : true
[INFO] includeLicenseText    : false
[INFO] outputReactorProjects : true
[INFO] outputFormat         : all
[INFO] outputName           : bom
[INFO] -----
[INFO] CycloneDX: Creating BOM
[INFO] CycloneDX: Writing BOM (XML): /home/sparsick/dev/workspace/maintenance-demo/target/
bom.xml
[INFO] CycloneDX: Validating BOM (XML): /home/sparsick/dev/workspace/maintenance-demo/target/
bom.xml
[INFO] CycloneDX: Writing BOM (JSON): /home/sparsick/dev/workspace/maintenance-demo/target/
bom.json
[INFO] CycloneDX: Validating BOM (JSON): /home/sparsick/dev/workspace/maintenance-demo/target/
bom.json
[WARNING] Unknown keyword additionalItems - you should define your own Meta Schema. If the
keyword is irrelevant for validation, just use a NonValidationKeyword
[INFO]
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.348 s
[INFO] Finished at: 2022-12-12T19:54:08+01:00
[INFO] -----
```



Wie können wir
am besten up to
date bleiben?



Dependabot

Automated dependency updates built into GitHub

As 593 followers

📍 United States of America

🌐 <https://github.com/features/security>

✉ support@github.com

✓ Verified

🏠 Overview 📁 Repositories 31 📁 Projects 📁 Packages 👤 People 11

Pinned

📁 [dependabot-core](#) Public

🔗 Dependabot's update PR creation logic. Feedback: <https://github.com/orgs/community/discussions/14691>

🔴 Ruby 🌟 1.1k 📄 774

📁 Repositories

🔍 Find a repository...

Type -

Language -

Sort -

[dependabot-core](#) Public

🔗 Dependabot's update PR creation logic. Feedback: <https://github.com/orgs/community/discussions/14691>



🔴 Ruby 🌟 1,062 📄 774 🕒 345.0k issues need help 📄 101 Updated 5 hours ago

[dependabot-actions-workflow](#) Public

Example workflow for updating Dependabot pull requests.



🔴 Ruby 🌟 21 📄 16 🕒 0 📄 2 Updated 9 hours ago

[dependabot-script](#) Public

A simple script that demonstrates how to use Dependabot Core.



🔴 Ruby 🌟 420 📄 N/A 📄 234 🕒 24 📄 10 Updated 5 days ago

[cli](#) Public

A tool for testing and debugging Dependabot update jobs.



🟢 Go 🌟 22 📄 N/A 📄 2 🕒 8 📄 2 Updated 6 days ago



MEND
Renovate

Supported Platforms

Renovate works on these platforms:

- [GitHub \(.com and Enterprise Server\)](#)
- [GitLab \(.com and CE/EE\)](#)
- [Bitbucket Cloud](#)
- [Bitbucket Server](#)
- [Azure DevOps](#)
- [AWS CodeCommit](#)
- [Gitea](#)

Label issues and pull requests for new contributors

[Dismiss](#)

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with **good first issue**

[Go to Labels](#)

Filters

Labels 7

Milestones 0

New pull request

☐ 13 Open ✓ 149 Closed

Author

Label

Projects

Milestones

Reviews

Assignee

Sort

☐ Update dependency org.apache.maven.plugins:maven-project-info-reports-plugin to v3.4.1 ✓

#310 opened 25 days ago by renovate: bot 1 task

☐ Update dependency org.apache.maven.plugins:maven-jxr-plugin to v3.3.0 ✓

#309 opened 26 days ago by renovate: bot 1 task

☐ Update dependency org.apache.maven.plugins:maven-javadoc-plugin to v3.4.1 ✓

#208 opened 26 days ago by renovate: bot 1 task

1

☐ Update dependency org.eclipse.platform:org.eclipse.osgi to v3.18.200 ✓

#292 opened on May 31 by renovate: bot 1 task

☐ Update dependency org.codehaus.plexus:plexus-component-metadata to v2.1.1 ✓

#291 opened on May 31 by renovate: bot 1 task

1

☐ Update dependency org.codehaus.plexus:plexus-component-annotations to v2.1.1 ✓

#290 opened on May 31 by renovate: bot 1 task

1

☐ Update dependency org.apache.maven.plugins:maven-plugin-plugin to v3.7.0 ✓

#284 opened on Dec 16, 2021 by renovate: bot 1 task

1

☐ Update dependency org.apache.maven.plugin-tools:maven-plugin-annotations to v3.7.0 ✓

#265 opened on Apr 26, 2021 by renovate: bot 1 task

1

Update dependency org.apache.maven.plugins:maven-project-info-reports-plugin to v3.4.1 #310

[Edit](#)[Code](#)[Gist](#)

renovate wants to merge 1 commit into [master](#) from [renovate/org.apache.maven.plugins:maven-project-info-reports-plugin-3.4.1](#)

[Conversation](#)[Commits](#)[Checks](#)[Files changed](#)[+1 -1](#)

renovate bot commented 25 days ago

[Contributor](#)

This PR contains the following updates:

Package	Change	Age	Adoption	Passing	Confidence
org.apache.maven.plugins:maven-project-info-reports-plugin	3.1.1 -> 3.4.1				

Configuration

Schedule: Branch creation - At any time (no schedule defined), Automerge - At any time (no schedule defined).

Automerge: Disabled by config. Please merge this manually once you are satisfied.

Rebasing: Whenever PR becomes conflicted, or you tick the rebase/retry checkbox.

Ignore: Close this PR and you won't be reminded about this update again.

☐ If you want to rebase/retry this PR, check this box

This PR has been generated by [Mend Renovate](#). View repository job log [here](#).

[Reviewers](#)[Suggestions](#)[Request](#)

Still in progress? Convert to draft.

[Assignees](#)

No one—assign yourself

[Labels](#)

None yet

[Projects](#)

None yet

[Milestone](#)

No milestone

[Development](#)

Successfully merging this pull request may close these issues.

None yet.

[Notifications](#)[Customize](#)[Unsubscribe](#)

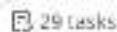
You're receiving notifications because you're watching this repository.

[Update dependency org.apache.maven.plugins:maven-project-info-reports-](#)[verified](#)[13bcd79](#)

Dependency Dashboard #278

[Edit](#)[New issue](#)[Gipad](#)

Open



29 tasks

renovate



opened this issue on Sep 14, 2021 · 0 comments



renovate bot commented on Sep 14, 2021 · edited

Contributor



This issue lists Renovate updates and detected dependencies. Read the [Dependency Dashboard](#) docs to learn more.

Rate-Limited

These updates are currently rate limited. Click on a checkbox below to force their creation now.

- ☐ Update dependency org.apache.maven.plugins:maven-site-plugin to v3.12.1
- ☐ Update dependency org.apache.maven.wagon:wagon-ftp to v3.5.2
- ☐ Update dependency org.apache.maven.wagon:wagon-ssh to v3.5.2
- ☐ Update dependency org.apache.rat:apache-rat-plugin to v0.15
- ☐ Update dependency org.assertj:assertj-core to v3.23.1
- ☐ Update dependency org.twdata.maven:mojo-executor to v2.4.0
- ☐ Update maven.version to v3.8.6 (org.apache.maven:maven-core , org.apache.maven:maven-artifact , org.apache.maven:maven-plugin-api , org.apache.maven:maven-model , org.apache.maven:maven-settings , org.apache.maven:maven-compat)
- ☐ Update release-drafter/release-drafter action to v5.21.1
- ☐ Update actions/cache action to v3
- ☐ Update actions/checkout action to v3
- ☐ Update actions/setup-java action to v3
- ☐ Update codecov/codecov-action action to v3
- ☐ Update dependency biz.aQute.bnd:biz.aQute.bndlib to v6
- ☐ Update dependency org.apache.maven.plugins:maven-gpg-plugin to v3
- ☐ Update dependency org.mockito:mockito-inline to v4
- ☐ Update tycho.version to v3 (major) (org.eclipse.tycho:sisu-equinox-launching , org.eclipse.tycho:tycho-core)
- ☐ Create all rate-limited PRs at once

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

[Create a branch](#) for this issue or link a pull request.

Notifications

[Customize](#)

Unsubscribe

You're receiving notifications because you're watching this repository.

0 participants

Lock conversation

Pin issue ⓘ

Transfer issue

Self Hosting

- Fertige GitHub Action und GitLab Runner
- Docker Image
- Mend Renovate On-Premises
- Mend Remediate (Kommerziell)



Renovate Global Config

\$ cat ./renovate/config.js

```
module.exports = {  
  repositories: ["sparsick/maintenance-demo"],  
  dryRun : "true",  
  prCommitsPerRunLimit : "5"  
};
```

Renovate Per Repository Config

\$ cat renovate.json

```
{  
  "extends": [  
    "config:base"  
  ]  
}
```



```
$ docker run --rm -v "${PWD}/.renovate/config.js:/usr/src/app/config.js" -e RENOVATE_TOKEN=apitoken renovate/renovate
```

```
INFO: Repository started (repository=sparsick/maintenance-demo)
      "renovateVersion": "34.55.0"
INFO: Dependency extraction complete (repository=sparsick/maintenance-demo, baseBranch=master)
      "stats": {
        "managers": {
          "docker-compose": {"fileCount": 1, "depCount": 1},
          "maven": {"fileCount": 1, "depCount": 16}
        },
        "total": {"fileCount": 2, "depCount": 17}
      }
INFO: DRY-RUN: Would commit files to branch renovate/org.asciidoctor-asciidoctorj-diagram-1.x (repository=sparsick/maintenance-demo,
branch=renovate/org.asciidoctor-asciidoctorj-diagram-1.x)
...
INFO: DRY-RUN: Would ensure Dependency Dashboard (repository=sparsick/maintenance-demo)
      "title": "Dependency Dashboard"
INFO: DRY-RUN: Would save repository cache. (repository=sparsick/maintenance-demo)
INFO: Repository finished (repository=sparsick/maintenance-demo)
      "cloned": true,
      "durationMs": 52013
```

Was ist bei Updates, wo ich viel Code anpassen
muss/will?

Java 8 → Java 11
JUnit4 → JUnit5



Large-Scale Automated Source Code Refactoring



```
<plugin>
  <groupId>org.openrewrite.maven</groupId>
  <artifactId>rewrite-maven-plugin</artifactId>
  <version>4.38.0</version>
  <configuration>
    <activeRecipes>
      <recipe>org.openrewrite.java.migrate.Java8toJava11</recipe>
    </activeRecipes>
  </configuration>
  <dependencies>
    <dependency>
      <groupId>org.openrewrite.recipe</groupId>
      <artifactId>rewrite-migrate-java</artifactId>
      <version>1.13.0</version>
    </dependency>
  </dependencies>
</plugin>
```

```

$ mvn rewrite:discover

[INFO] Searching for projects...
[INFO] ----- org.springframework.samples:spring-petclinic -----
[INFO] detecting petclinic 2.2.8.BUILD-SNAPSHOT
[INFO] ----- [ jar ] -----
[INFO] --- rewrite-avena-plugin:4.38.3:discover (default-cli) @ spring-petclinic ---
[INFO] Available Recipes:
....
[INFO]      org.openrewrite.java.cleanup.AddSerialVersionUIDToSerializable
[INFO]      org.openrewrite.java.cleanup.AstericPrimitiveEqualsUsesGet
[INFO]      org.openrewrite.java.format.EmptyNewLinesAtEndOfFile
[INFO]      org.openrewrite.java.format.NormalizeTabSizeSpaces
[INFO]      org.openrewrite.java.migrate.AddSuppressionForIllegalReflectionFindingsPlugin
[INFO]      org.openrewrite.java.migrate.Java8toJava11
[INFO]      org.openrewrite.java.migrate.UseJavaUtilBase64
[INFO]      org.openrewrite.java.migrate.apache.commons.codec.AsciiBase64ToBase64
[INFO]      org.openrewrite.java.migrate.quava.NaiveJava8Annotations
...
[INFO] Available Styles:
[INFO]      org.openrewrite.java.util
[INFO]      com.netflix.eureka.Style
[INFO]      org.openrewrite.java.docx.javaFormat
[INFO]      org.openrewrite.java.SpringFormat
[INFO]      com.netflix.genie.Style
[INFO]
[INFO] Active Styles:
[INFO]
[INFO] Active Recipes:
[INFO]      org.openrewrite.java.migrate.Java8toJava11
[INFO]
[INFO] Found 416 available recipes and 5 available styles.
[INFO] Configured with 1 active recipes and 0 active styles.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 4.708 s
[INFO] Finished at: 2022-12-13T18:17:34+01:00
[INFO]

```

Beispiel: Java 8 nach Java 11

```
<plugin>
  <groupId>org.openrewrite.maven</groupId>
  <artifactId>rewrite-maven-plugins</artifactId>
  <version>4.38.0</version>
  <configuration>
    <activeRecipes>
      <recipe>org.openrewrite.java.migrate.Java8toJava11</recipe>
    </activeRecipes>
  </configuration>
  <dependencies>
    <dependency>
      <groupId>org.openrewrite.recipe</groupId>
      <artifactId>rewrite-migrate-java</artifactId>
      <version>1.13.0</version>
    </dependency>
  </dependencies>
</plugin>
```

```
$ mvn rewrite:run
[INFO] --- rewrite-maven-plugin:4.38.0:run (default-cli) @ spring-petclinic ---
[INFO] Using active recipe(s) [org.openrewrite.java.migrate.Java8toJava11]
[INFO] Using active styles(s) []
[INFO] Validating active recipes...
[INFO] Project [petclinic] Resolving Poms...
[INFO] Project [petclinic] Parsing Source Files
[INFO] Running recipe(s)...
[WARNING] Changes have been made to pom.xml by:
[WARNING]     org.openrewrite.java.migrate.Java8toJava11
[WARNING]         org.openrewrite.java.migrate.javax.AddJaxbDependencies
[WARNING]             org.openrewrite.java.migrate.javax.AddJaxbRuntime: {runtime=sun}
[WARNING]             org.openrewrite.java.migrate.AddJDepSrcPlugin: {release=}
[WARNING]             org.openrewrite.java.migrate.wro4j.UpgradeWro4jMavenPluginVersion
[WARNING]             org.openrewrite.maven.UpgradePluginVersion: {groupId=ro.isdc.wro4j,
artifactId=wro4j-maven-plugin, newVersion=1.10.1}
[WARNING]             org.openrewrite.java.migrate.jacoco.UpgradeJaCoCoMavenPluginVersion
[WARNING]             org.openrewrite.maven.UpgradePluginVersion: {groupId=org.jacoco,
artifactId=jacoco-maven-plugin, newVersion=0.8.8}
[WARNING]             org.openrewrite.java.migrate.javaVersion[]
[WARNING]             org.openrewrite.maven.ChangePropertyValue: {key=java.version, newValue=11,
addIfMissing=false}
[WARNING] Please review and commit the results.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 34.463 s
[INFO] Finished at: 2022-12-13T11:07:24+01:00
[INFO] -----
```



```
$ git diff
diff --git a/pom.xml b/pom.xml
index 75c36f9..ff039b4 100644
--- a/pom.xml
+++ b/pom.xml
@@ -17,7 +17,7 @@
 <properties>

     <!-- Generic properties -->
-    <java.version>1.8</java.version>
-    <java.version11>11</java.version11>
+    <java.version>1.8</java.version>
+    <java.version11>11</java.version11>

     <!-- Spring versions -->
-    <spring.version>5.2.2</spring.version>
-    <spring.version11>5.2.2</spring.version11>
+    <spring.version>5.2.2</spring.version>
+    <spring.version11>5.2.2</spring.version11>

@@ -91,8 +91,10 @@
     <groupId>org.springframework</groupId>
     <artifactId>spring-context</artifactId>
 </dependency>
+<dependencies>
+<dependency>
     <groupId>org.springframework</groupId>
     <artifactId>spring-context</artifactId>
     <version>5.2.2</version>
     <scope>provided</scope>
 </dependency>
+<dependency>
     <groupId>org.springframework</groupId>
     <artifactId>spring-context</artifactId>
     <version>5.2.2</version>
     <scope>provided</scope>
 </dependency>
+<dependency>
     <groupId>org.springframework</groupId>
     <artifactId>spring-context</artifactId>
@@ -238,59 +238,57 @@
 </dependency>
 </dependencies>
 </parent>
</project>

```

Beispiel: Java 11 nach Java 17



```
$ mvn rewrite:discover | grep Java17  
[INFO]                org.openrewrite.java.migrate.UpgradeJava17
```


java rewrite:run

```
[INFO] --- rewrite-maven-plugin:4.38.0:run (default-cli) @ spring-petclinic ---
[INFO] Using active recipe(s) [org.openrewrite.java.migrate.UpgradeJava17]
[INFO] Using active style(s) []
[INFO] Validating active recipes...
[INFO] Project [petclinic] Resolving Poms...
[INFO] Project [petclinic] Parsing Source Files
[INFO] Running recipe(s)...
[WARNING] Changes have been made to pom.xml by:
[WARNING]     org.openrewrite.java.migrate.UpgradeJava17
[WARNING]     org.openrewrite.java.migrate.Java8toJava11
[WARNING]     org.openrewrite.java.migrate.javax.AddJaxbDependencies
[WARNING]     org.openrewrite.java.migrate.javax.AddJaxbRuntime: (runtime=son)
[WARNING]     org.openrewrite.java.migrate.wro4j.UpgradeWro4jMavenPluginVersion
[WARNING]     org.openrewrite.maven.UpgradePluginVersion: {groupId=ro.isdc.wro4j, artifactId=wro4j-maven-plugin,
newVersion=1.9.1}
[WARNING]     org.openrewrite.java.migrate.JavaVersion11
[WARNING]     org.openrewrite.maven.ChangePropertyValue: {key=java.version, newValue=11, addIfMissing=false}
[WARNING]     org.openrewrite.java.migrate.JavaVersion17
[WARNING]     org.openrewrite.maven.ChangePropertyValue: {key=java.version, newValue=17, addIfMissing=false}
[WARNING] Changes have been made to src/test/java/org/springframework/samples/petclinic/owner/OwnerControllerTests.java by:
[WARNING]     org.openrewrite.java.migrate.UpgradeJava17
[WARNING]     org.openrewrite.java.migrate.Java8toJava11
[WARNING]     org.openrewrite.java.migrate.util.JavaUtilAPIs
[WARNING]     org.openrewrite.java.migrate.util.MigrateCollectionsSingletonList
[WARNING]     org.openrewrite.java.migrate.util.MigrateCollectionsSingletonSet
[WARNING] Please review and commit the results.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 32.266 s
[INFO] Finished at: 2022-12-13T11:46:21+01:00
[INFO] -----
```


\$ 2.00.

```
$ sdk use java 17.0.0-aa
$ mvn clean verify
```

```
...
[INFO] Running org.springframework.samples.petclinic.service.ClinicServiceTests
[ERROR] Tests run: 1, Failures: 0, Errors: 1, Skipped: 0, Time elapsed: 0 s <= FAILURE! - in
org.springframework.samples.petclinic.service.ClinicServiceTests
[ERROR] InitializationError Time elapsed: 0 s <= ERROR!
org.springframework.beans.factory.BeanDefinitionStoreException: Failed to read candidate component class: file [/home/sparsick/
/dev/workspace/maintenance-demo/target/test-classes/org/springframework/samples/petclinic/service/ClinicServiceTests.class]; nested
exception is org.springframework.core.NestedIOException: ASM ClassReader failed to parse class file - probably due to a new Java
class file version that isn't supported yet: file [/home/sparsick/dev/workspace/maintenance-demo/target/test-classes
/org/springframework/samples/petclinic/service/ClinicServiceTests.class]; nested exception is java.lang.IllegalArgumentException:
Unsupported class file major version 61
Caused by: org.springframework.core.NestedIOException: ASM ClassReader failed to parse class file - probably due to a new Java class
file version that isn't supported yet: file [/home/sparsick/dev/workspace/maintenance-demo/target/test-classes/org/springframework
/samples/petclinic/service/ClinicServiceTests.class]; nested exception is java.lang.IllegalArgumentException: Unsupported class file
major version 61
Caused by: java.lang.IllegalArgumentException: Unsupported class file major version 61

[INFO] Running org.springframework.samples.petclinic.PetClinicIntegrationTests
[ERROR] Tests run: 1, Failures: 0, Errors: 1, Skipped: 0, Time elapsed: 0 s <= FAILURE! - in
org.springframework.samples.petclinic.PetClinicIntegrationTests
[ERROR] InitializationError Time elapsed: 0 s <= ERROR!
org.springframework.beans.factory.BeanDefinitionStoreException: Failed to read candidate component class: file [/home/sparsick/
/dev/workspace/maintenance-demo/target/test-classes/org/springframework/samples/petclinic/PetClinicIntegrationTests.class]; nested
exception is org.springframework.core.NestedIOException: ASM ClassReader failed to parse class file - probably due to a new Java
class file version that isn't supported yet: file [/home/sparsick/dev/workspace/maintenance-demo/target/test-classes
/org/springframework/samples/petclinic/PetClinicIntegrationTests.class]; nested exception is java.lang.IllegalArgumentException:
Unsupported class file major version 61
Caused by: org.springframework.core.NestedIOException: ASM ClassReader failed to parse class file - probably due to a new Java class
file version that isn't supported yet: file [/home/sparsick/dev/workspace/maintenance-demo/target/test-classes/org/springframework
/samples/petclinic/PetClinicIntegrationTests.class]; nested exception is java.lang.IllegalArgumentException: Unsupported class file
major version 61
Caused by: java.lang.IllegalArgumentException: Unsupported class file major version 61
...

```

Beispiel: JUnit4 nach JUnit5

```
<plugin>
  <groupId>org.openrewrite.maven</groupId>
  <artifactId>rewrite-maven-plugin</artifactId>
  <version>4.38.0</version>
  <configuration>
    <activeRecipes>
      <recipe>org.openrewrite.java.spring.boot2.SpringBoot2JUnit4to5Migration</recipe>
    </activeRecipes>
  </configuration>
  <dependencies>
    <dependency>
      <groupId>org.openrewrite.recipe</groupId>
      <artifactId>rewrite-spring</artifactId>
      <version>4.29.0</version>
    </dependency>
  </dependencies>
</plugin>
```

```

$ mvn compile
[INFO] Scanning for projects...
[INFO] Using active profiles[]
[INFO] Initializing project modules...
[INFO] Project [petclinic:petclinic] resolving...
[INFO] Project [petclinic:petclinic] resolving...
[INFO] Building petclinic...
[INFO] Changes have been made to pom.xml file
[INFO] org.springframework.java.spring:core2.SpringBoot2Unit4to9Integration
[INFO] org.springframework.java.test:unit4.Unit4to9Integration
[INFO] org.springframework.maven.ExcludedDependency: (org.junit:junit, artifactId:junit)
[INFO] org.springframework.maven.ExcludedDependency: (org.springframework:spring-test:spring-test)
[INFO] Changes have been made to src/test/java/org.springframework.samples.petclinic.crewer/Unit4to9IntegrationTests.java by
[INFO] org.springframework.java.spring:core2.SpringBoot2Unit4to9Integration
[INFO] org.springframework.java.test:unit4.Unit4to9Integration
[INFO] org.springframework.java.test:unit5.UpdatedTestAnnotation
[INFO] org.springframework.java.test:unit5.UpdatedTestAnnotation
[INFO] org.springframework.java.test:unit5.RemoteTestExtension: framework=
[INFO] org.springframework.test:unit4.SpringRunner, org.springframework.test:unit4.SpringRunner, junit4.SpringUnit4ClassRunner,
[INFO] org.springframework.test:unit4.SpringRunner, junit4.SpringRunner, junit4.SpringRunner, junit4.SpringRunner, junit4.SpringRunner,
[INFO] org.springframework.java.test:unit5.UpdatedTestAnnotation
[INFO] Changes have been made to src/test/java/org.springframework.samples.petclinic.crewer/Unit4to9IntegrationTests.java by
[INFO] org.springframework.java.spring:core2.SpringBoot2Unit4to9Integration
[INFO] org.springframework.java.test:unit4.Unit4to9Integration
[INFO] org.springframework.java.test:unit5.UpdatedTestAnnotation
[INFO] org.springframework.java.test:unit5.UpdatedTestAnnotation
[INFO] org.springframework.java.test:unit5.RemoteTestExtension: framework=
[INFO] org.springframework.test:unit4.SpringRunner, org.springframework.test:unit4.SpringRunner, junit4.SpringUnit4ClassRunner,
[INFO] org.springframework.test:unit4.SpringRunner, junit4.SpringRunner, junit4.SpringRunner, junit4.SpringRunner, junit4.SpringRunner,

```

[illegible]

• 918 d. 11

[illegible]

Wie vermeide ich, dass „alter Style“ wieder in meinen Code reinkommt?



OVERVIEW

Introduction

Goals

Usage

FAQ

License

Download

Releases History

EXAMPLES

Upgrading Checkstyle at Runtime

Using an Inline Checkstyle Checker Configuration

Using a Custom Checkstyle Checker Configuration

Using Custom Checkstyle Property Expansion Definitions

Using a Suppressions Filter

Using Custom Developed Checkstyle Checks

Multimodule Configuration

PROJECT DOCUMENTATION

Project Information

Apache Maven Checkstyle Plugin

The Checkstyle Plugin generates a report regarding the code style used by the developers. For more information, see the [FAQ](#).

This version of the plugin uses Checkstyle 9.3 by default and requires Java 8. But you can [upgrade](#) to a newer version.

The plugin can be configured in the project's POM. Predefined rulesets available for use in this version are listed in the [FAQ](#).

Goals Overview

The Checkstyle Plugin has three goals:

- `checkstyle:checkstyle` is a reporting goal that performs Checkstyle analysis and generates a report.
- `checkstyle:checkstyle-aggregate` is a reporting goal that performs Checkstyle analysis and generates a report for the entire project.
- `checkstyle:check` is a goal that performs Checkstyle analysis and outputs violations or a count.

Major Version Upgrade to version 3.0.0

Please note that the following parameters have been completely removed from the plugin configuration:

- `sourceDirectory` : use `sourceDirectories` instead;
- `testSourceDirectory` : use `testSourceDirectories` instead.

Usage

General instructions on how to use the Checkstyle Plugin can be found on the [usage page](#). Some

```
if-({  
  .!spotless  
  " " }- {  
  ..dirty();  
  ..}  
}
```

Spotless: Keep your code spotless with Maven

mavencentral: [com.diffplug.spotless:spotless-maven-plugin](#) [javadoc: yes](#) [changelog: 2.28.0](#)

[circleci: passing](#) [gitlab: chat](#) [license: apache](#)

Spotless is a general-purpose formatting plugin used by [4,000 projects on GitHub \(August 2020\)](#). It is completely à la carte, but also includes powerful "batteries-included" if you opt-in. Plugin requires a version of Maven higher or equal to 3.1.0.

To people who use your build, it looks like this:

```
user@machine repo % mvn spotless:check  
[ERROR] > the following files had format violations:  
[ERROR] src\main\java\com\diffplug\gradle\spotless\FormatExtension.java  
[ERROR] -\t\t---if-(targets.length==0){  
[ERROR] +\t\tif-(targets.length==0){  
[ERROR] Run 'mvn spotless:apply' to fix these violations.  
user@machine repo % mvn spotless:apply  
[INFO] BUILD SUCCESS  
user@machine repo % mvn spotless:check  
[INFO] BUILD SUCCESS
```



Modernizer Maven Plugin

maven-central v2.5.0

Modernizer Maven Plugin detects uses of legacy APIs which modern Java versions supersede. These modern APIs are often more performant, safer, and idiomatic than the legacy equivalents. For example, Modernizer can detect uses of `Vector` instead of `ArrayList`, `String.getBytes(String)` instead of `String.getBytes(Charset)`, and Guava `Objects.equal` instead of Java 7 `Objects.equals`. The default configuration detects [over 200 legacy APIs](#), including third-party libraries like [Apache Commons](#), [Guava](#), and [Joda-Time](#).

Configuration

To run Modernizer, add the following to the `<plugins>` stanza in your `pom.xml` then invoke `mvn modernizer:modernizer`:

```
<plugin>
  <groupId>org.gaul</groupId>
  <artifactId>modernizer-maven-plugin</artifactId>
  <version>2.4.0</version>
  <configuration>
    <javaVersion>8</javaVersion>
  </configuration>
</plugin>
```

Policeman's Forbidden API Checker

Allows to parse Java byte code to find invocations of method/class/field signatures and fail build (Apache Ant, Apache Maven, or Gradle).

maven-central v3.4 build passing

Documentation

Please refer to the Github [Wiki & Documentation](#).

The checker is available as Apache Ant Task, Apache Maven Mojo, and Gradle plugin. In addition there is a command line tool (CLI):

- [Apache Ant](#)
- [Apache Maven](#)
- [Gradle](#)
- [Command Line](#)

This project uses Apache Ant (and Apache Ivy) to build. The minimum Ant version is 1.8.0 and it is recommended to not have Apache Ivy in the Ant lib folder, because the build script will download the correct version of Ivy automatically.

Was ist mit Abhängigkeiten, die nicht mehr genutzt werden?



DepClean



What is DepClean?

DepClean automatically cleans the dependency tree of Java projects. It removes all the dependencies that are included in the project's dependency tree but are not actually necessary to build it. DepClean detects and removes all the unused dependencies declared in the `pom.xml` file of a project or imported from its parent. It can be executed as a Maven goal through the command line or integrated directly into the Maven build lifecycle (CI/CD). DepClean does not modify the original source code of the application nor its original `pom.xml`.

For a visual illustration of what DepClean can provide for your project, have a look at the companion [dep-clean-web](#) project.


```

$ mvn condition:analyze
$ mvn se:ath,cas:princepclean-neven-plugins:2.0.3:depclean -DresultPath=depclean -DresultResult=2.0.3

[INFO] --- depclean-neven-plugins:2.0.3:depclean (default-cli) @ spring-petclinic ---

[INFO] Starting DepClean dependency analysis

=====
REPORT FROM ANALYSIS RESULTS
=====

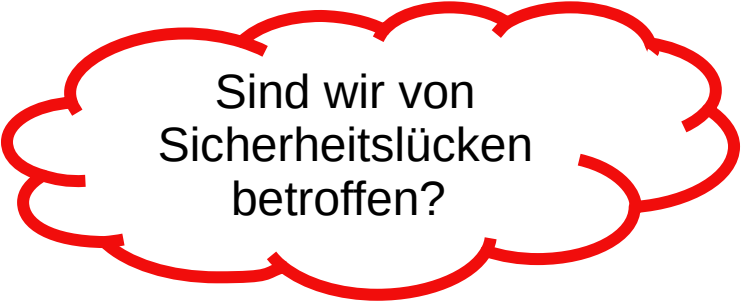
USED DIRECT DEPENDENCIES [7]:
...
USED INHERITED DEPENDENCIES [0]:
USED TRANSITIVE DEPENDENCIES [44]:
...
POTENTIALLY UNUSED DIRECT DEPENDENCIES [12]:
  org.springframework:3.8.0:compile (*) PR
  org.webjars.bootstrap:3.3.6:compile (646 KB)
  org.webjars.jquery-misc:1.1.4:compile (251 KB)
  org.webjars.jquery:2.2.4:compile (158 KB)
  commons-collections:3.1.1:compile (152 KB)
  org.springframework:spring-test:4.3.1:test (520 bytes)
  org.springframework.boot:spring-boot-starter-data-jpa:2.2.6.RELEASE:compile (1400 bytes)
  org.springframework.boot:spring-boot-starter-thymeleaf:2.2.6.RELEASE:compile (1400 bytes)
  org.springframework.boot:spring-boot-starter-web-services:2.2.6.RELEASE:compile (1400 bytes)
  org.springframework.boot:spring-boot-starter-cache:2.2.6.RELEASE:compile (434 bytes)
  org.springframework.boot:spring-boot-starter-web:2.2.6.RELEASE:compile (480 bytes)
  org.springframework.boot:spring-boot-starter-test:2.2.6.RELEASE:test (482 bytes)

POTENTIALLY UNUSED INHERITED DEPENDENCIES [6]:
POTENTIALLY UNUSED TRANSITIVE DEPENDENCIES [20]:
...
[INFO] Starting depcleaning task
[INFO] Adding 26 unused transitive dependencies as direct dependencies.
[INFO] Removing 12 unused direct dependencies.
[INFO] Excluding 20 unused transitive dependencies one-by-one.
[INFO] POM debloated successfully.
[INFO] pom-debloated.xml file created in: /home/sparstick/dev/workspace/maintenance-dono/pom-debloated.xml.
[INFO] Creating depclean-results.json, please wait...
[INFO] depclean-results.json file created in: /home/sparstick/dev/workspace/maintenance-dono/target/depclean-results.json
[INFO] Analysis done in 8min 12s
[INFO] -----
[INFO] BUILD SUCCESS[INFO]
[INFO] Total time: 11.59M s
[INFO] Finished at: 2022-12-17T15:22:43+01:00
[INFO]

```


Visualisierung via depclean-web

spring-petclinic 2.2.0.BUILD-SNAPSHOT				
dependencies	direct	transitive	provided	
Used	19	102	0	DepClean Project
Unused	12	28	0	



Sind wir von
Sicherheitslücken
betroffen?



OWASP [Dependency-Check](#) [documentation](#) / [About](#)

OWASP DEPENDENCY-CHECK

General

- [How It Works](#)
- [Reading the Report](#)
- [False Positives](#)
- [False Negatives](#)
- [Internet Access Required](#)
- [Related Work](#)
- [Project Presentation \(pptx\)](#)
- [Project Presentation \(pdf\)](#)
- [Sample Report](#)
- [How to Scan an ISO Image](#)

File Type Analyzers

Modules

PROJECT DOCUMENTATION

Project Information

Project Reports

[Follow @owasp](#) 1,062 followers



About

OWASP dependency-check is an open source solution to the OWASP Top 10 2021 entry: [/ Analyzer](#) page). Note that some of the analyzers are experimental and may produce more

The problem with using known vulnerable components was covered in a paper by Jeff Will vulnerabilities (such as those at the [National Vulnerability Database](#)).

More information about dependency-check can be found [here](#):

- [How does dependency-check work](#)
- [How to read the report](#)

OWASP dependency-check's core analysis engine can be used as:

- [Ant Task](#)
- [Command Line Tool](#)
- [Gradle Plugin](#)
- [Jenkins Plugin](#)
- [Maven Plugin](#) - Maven 3.1 or newer required
- [SBT Plugin](#)

For help with dependency-check the following resource can be used:

- [Open a github issue](#)

Sponsor Development of dependency-check:





```
<plugin>  
  <groupId>org.owasp</groupId>  
  <artifactId>dependency-check-maven</artifactId>  
  <version>7.4.1</version>  
</plugin>
```

```
$ mvn dependency-check:check
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.springframework.samples:spring-petclinic >-----
[INFO] Building petclinic 2.2.0.BUILD-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- dependency-check-maven:7.4.1:check (default-cli) @ spring-petclinic ---
[INFO] Checking for updates
[INFO] Skipping NVD check since last check was within 4 hours.
[INFO] Skipping RetireJS update since last update was within 24 hours.
[INFO] Check for updates complete (19 ms)
[INFO]
```

Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the analysis or its use. Any use of the tool and the reporting provided is at the user's risk. In no event shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

About DDC: <https://jeremylong.github.io/DependencyCheck/general/internals.html>

False Positives: <https://jeremylong.github.io/DependencyCheck/general/suppression.html>

🍷 Sponsor: <https://github.com/sponsors/jeremylong>

```

[INFO] Analysis Started
[INFO] Finished Archive Analyzer (8 seconds)
[INFO] Finished File Name Analyzer (8 seconds)
[INFO] Finished Jar Analyzer (0 seconds)
[WARNING] No lock file exists - this will result in false negatives; please run 'npm install --package-lock'
[WARNING] No lock file exists - this will result in false negatives; please run 'npm install --package-lock'
[WARNING] Analyzing '/tmp/dt1ea73eeaf8-5615-4323-8aac-2f89ad491c25/check15701359662366963867tmp/64/MFT4-TRF/resources/webpack/
/angular http/2.4.10/package.json' - however, the node_modules directory does not exist. Please run 'npm install' prior to running
dependency-check.
[WARNING] Analyzing '/tmp/dt1ea73eeaf8-5615-4323-8aac-2f89ad491c25/check15701359662366963867tmp/64/MFT4-TRF/resources/webpack/
/angular http/2.4.10/testing/package.json' - however, the node_modules directory does not exist. Please run 'npm install' prior to
running dependency-check.
[INFO] Finished Node.js Package Analyzer (0 seconds)
[INFO] Finished Dependency Berging Analyzer (8 seconds)
[INFO] Finished Version Filter Analyzer (0 seconds)
[INFO] Finished Hint Analyzer (0 seconds)
[INFO] Created CVE Index (1 seconds)
[INFO] Finished CVE Analyzer (13 seconds)
[INFO] Finished False Positive Analyzer (0 seconds)
[INFO] Finished EVD CVE Analyzer (8 seconds)
[INFO] Finished RetireJS Analyzer (8 seconds)
[INFO] Finished Sonatype USS Index Analyzer (8 seconds)
[INFO] Finished Vulnerability Suppression Analyzer (0 seconds)
[INFO] Finished Dependency Bundling Analyzer (8 seconds)
[INFO] Finished Trusted Suppression Rule Analyzer (8 seconds)
[INFO] Analysis Complete (5 seconds)
[INFO] Writing report to: /home/sparsick/cveworkspace/maintenance-data/target/dependency-check-report.html
[WORKING]

```

One or more dependencies were identified with known vulnerabilities in petchline:

```

bootstrap-3.3.6.jar (pkg:[javascript/bootstrap]3.3.6, pkg:[maven/org.webjars/bootstrap]3.3.6) : CVE-2016-18735, CVE-2018-14040,
CVE-2018-14941, CVE-2018-14642, CVE-2018-20077, CVE-2019-8331

```

...

See the dependency-check report for more details.

```

[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 8.038 s
[INFO] Finished at: 2022-12-13T18:54:21+01:00
[INFO]

```




Dependency-Check is an open source tool performing a best effort analysis of 140+ popular dependencies. false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an ADI environment, and there are ADI environments implied or otherwise, with regard to the analysis or its use. Use of the tool and the reporting provided is the user's risk. Some users shall file copyright notice or DMCA/DFIR hold notice for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

[How to read the report](#) | [Suppressing false positives](#) | [Getting Help: github issues](#)

[Sponsor](#)

Project: petclinic

org.springframework.samples.spring-petclinic:2.2.0.BUILD-SNAPSHOT

Scan Information ([show all](#))

- dependency-check version: 7.4.1
- Report Generated On: Tue, 13 Dec 2022 13:54:20 +0100
- Dependencies Scanned: 161 (117 unique)
- Vulnerable Dependencies: 20
- Vulnerabilities Found: 106
- Vulnerabilities Suppressed: 0
- ...

Summary

Display: [Showing Vulnerable Dependencies \(click to show all\)](#)

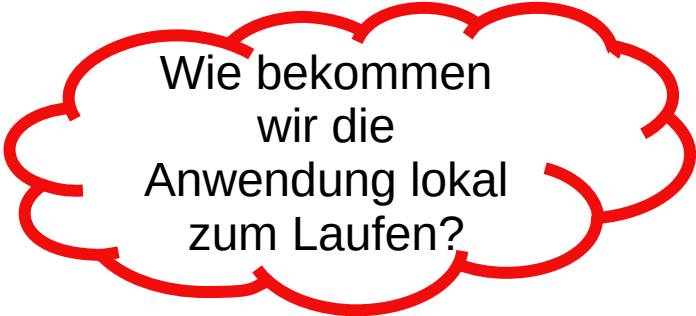
Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
bootstrap-3.3.6.jar		org.tycho.sprngbootstrap@3.3.6 org.springframework.web.servlet.bootstrap@3.3.6	MEDIUM	6		19
dom4j-2.1.1.jar	cpe:2.3:a:dom4j_project:dom4j:2.1.1:*****	org.springframework.dom4j:dom4j@2.1.1	CRITICAL	1	Highest	20
hibernate-core-5.4.6.Final.jar	cpe:2.3:a:hibernate:hibernate_core:5.4.6:*****	org.springframework.hibemate:hibernate-core@5.4.6.Final	HIGH	2	Low	48
hibernate-validator-6.0.17.Final.jar	cpe:2.3:a:redhat:hibernate_validator:6.0.17:*****	org.springframework.hibemate.validator:hibernate-validator@6.0.17.Final	MEDIUM	2	Highest	34
hsqldb-2.5.0.jar	cpe:2.3:a:hsqldb:hypersql_database:2.5.0:*****	org.springframework.hsqldb:hsqldb@2.5.0	CRITICAL	1	Low	41
jackson-databind-2.10.0.jar	cpe:2.3:a:fasterxml:jackson-databind:2.10.0:***** cpe:2.3:a:fasterxml:jackson-morphia:2.10.0:*****	org.springframework.fasterxml:jackson-core:jackson-databind@2.10.0	HIGH	4	Highest	41

Und was mit Docker Images?

```
→ trivy i --ignore-unfixed -o result spring-boot-demo:latest
2022-06-23T09:55:56.244+0200 INFO Vulnerability scanning is enabled
2022-06-23T09:55:56.245+0200 INFO Secret scanning is enabled
2022-06-23T09:55:56.245+0200 INFO If your scanning is slow, please try '--security-checks vuln'
to disable secret scanning
2022-06-23T09:55:56.245+0200 INFO Please see also https://aquasecurity.github.io/trivy/v0.29.2
/docs/secret/scanning/#recommendation for faster secret detection
2022-06-23T09:55:56.254+0200 INFO Detected OS: debian
2022-06-23T09:55:56.255+0200 INFO Detecting Debian vulnerabilities...
2022-06-23T09:55:56.265+0200 INFO Number of language-specific files: 1
2022-06-23T09:55:56.265+0200 INFO Detecting jar vulnerabilities...

spring-boot-demo:latest (debian 11.2)
=====
Total: 23 (UNKNOWN: 1, LOW: 2, MEDIUM: 6, HIGH: 6, CRITICAL: 8)

Java (jar)
=====
Total: 0 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 0, CRITICAL: 0)
```



Wie bekommen
wir die
Anwendung lokal
zum Laufen?



Spring Boot Maven Plugin



```
version: "3.9"
services:
  database:
    image: mongo:4.2.21
    restart: always
    ports:
      - 27017:27017
    environment:
      MONGO_INITDB_ROOT_USERNAME: root
      MONGO_INITDB_ROOT_PASSWORD: root123
    volumes:
      - ./local-env/./dockerentrypoint-initdb.d/
```



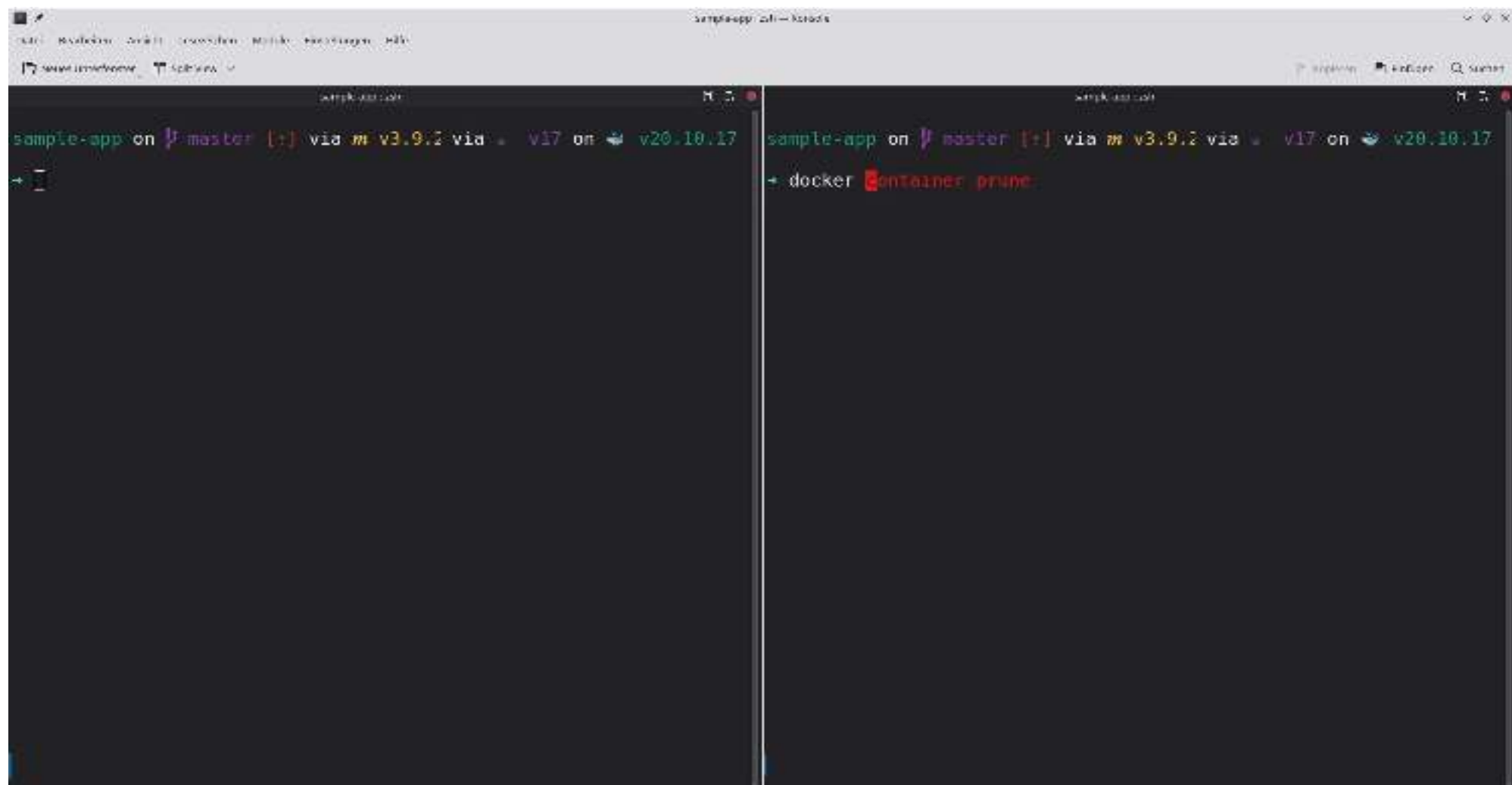
```
<plugin>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-maven-plugin</artifactId>
  <configuration>
    <layers>
      <enabled>true</enabled>
    </layers>
    <environmentVariables>
      <MONGODB_ENABLED>true</MONGODB_ENABLED>
      <MONGODB_URI>mongodb://test:test123@localhost/test</MONGODB_URI>
    </environmentVariables>
  </configuration>
</plugin>
```


Spring Boot Docker Compose Support

```
<!-- Since Spring Boot 3.1 -->
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-docker-compose</artifactId>
    <optional>true</optional>
  </dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <configuration>
        <layers>
          <enabled>true</enabled>
        </layers>
      </configuration>
    </plugin>
  </plugins>
</build>
```



```
# application.properties
# default: compose.yml
spring.docker.compose.file=docker-compose.yml
```



Onboarding neuer
Devs



Get started now!

Go on, paste and run the following in a terminal:

```
curl -s "https://get.sdkman.io" | bash
```

The Software Development Kit Manager

SDKMAN! is a tool for managing parallel versions of multiple **Software Development Kits** on most Unix-based systems. It provides a convenient Command Line Interface (CLI) and API for installing, switching, removing and listing Candidates. Formerly known as **GVM** the **Groovy Environment Manager**, it was inspired by the very useful **RVM** and **rvm** tools, used at large by the Ruby community.



By Developers, for Developers

Making life easier for those building, downloading, editing, organizing archives, dealing with **JDKs** and **RVM** environment variables.



Multi-platform

Runs on any UNIX-based platform: macOS, Linux, Cygwin, Solaris, and FreeBSD.



Java all the way down

Install software development kits for the JVM such as Java, Groovy, Scala, Kotlin and Clojure. Ant, Gradle, Grails, Maven, SBT, Spark, Spring Boot, Vert.x and many others also supported.



APIs

New clients can easily be written by consuming our open broker REST API. Vendors can publish and announce their own releases through a secure Vendor API.



Lightweight

Written in **bash** and only requires **curl** and **dpkg** to be present on your system. Does work with **SSL** too.



Open Source

Backed by a global community of developers.
Licensed under **Apache 2.0**.



```
$ sdk install java 17-open
$ sdk install maven 3.6.3
$ sdk install maven 3.8.3
$ sdk use java 17-open
$ sdk use maven 3.8.3
$ sdk env init
$ cat .sdkmanrc
# Enable auto-env through the sdkman_auto_env config
# Add key=value pairs of SDKs to use below
java=17-open
maven=3.8.3
```



EditorConfig

What is EditorConfig?

Example File

File Location

File Format Details

No Plugin Necessary

Download a Plugin

Contributing

Blog

Project Page
on GitHub



Follow Us
on Twitter



Reddit

What is EditorConfig?

EditorConfig helps maintain consistent coding styles for multiple developers working on the same project across various editors and IDEs. The EditorConfig project consists of a **file format** for defining coding styles and a collection of **text editor plugins** that enable editors to read the file format and adhere to defined styles. EditorConfig files are easily readable and they work nicely with version control systems.

What's an EditorConfig file look like?

(A [format specification of EditorConfig](#) is also available.)

Example file

Below is an example `.editorconfig` file setting end-of-line and indentation styles for Python and JavaScript files.

```
# EditorConfig is awesome:
https://EditorConfig.org

# top-most EditorConfig file
root = true

# Unix-style newlines with a newline ending every file
[*]
end_of_line = lf
insert_final_newline = true
```

Fazit

Bei Greenfield Projekte

Automatisierung
(was man automatisieren kann)
von Beginn an

Bei Brownfield Projekte

Guckt was die größten Schmerzen bereiten und versucht diese weg zu automatisieren.

Ähm, und was ist mit AI?

Code maintenance



Second

Fragen?

mail@sandra-parsick.de

@SandraParsick

@sparsick@mastodon.social

<https://github.com/sparsick/maintenance-talk>

Referenzen

- <https://github.com/spring-projects/spring-petclinic>
- <https://spring-petclinic.github.io/>
- <https://www.arc42.de/overview/>
- <https://adr.github.io/>
- <https://www.unusual.vc/post/ai-software-development>

Bildnachweise

- https://pxhere.com/de/photo/970825?utm_content=shareClip&utm_medium=referral&utm_source=pxhere
- https://pxhere.com/de/photo/867723?utm_content=shareClip&utm_medium=referral&utm_source=pxhere
- https://pxhere.com/de/photo/1409158?utm_content=shareClip&utm_medium=referral&utm_source=pxhere
- <https://www.pexels.com/de-de/foto/hausronovio>