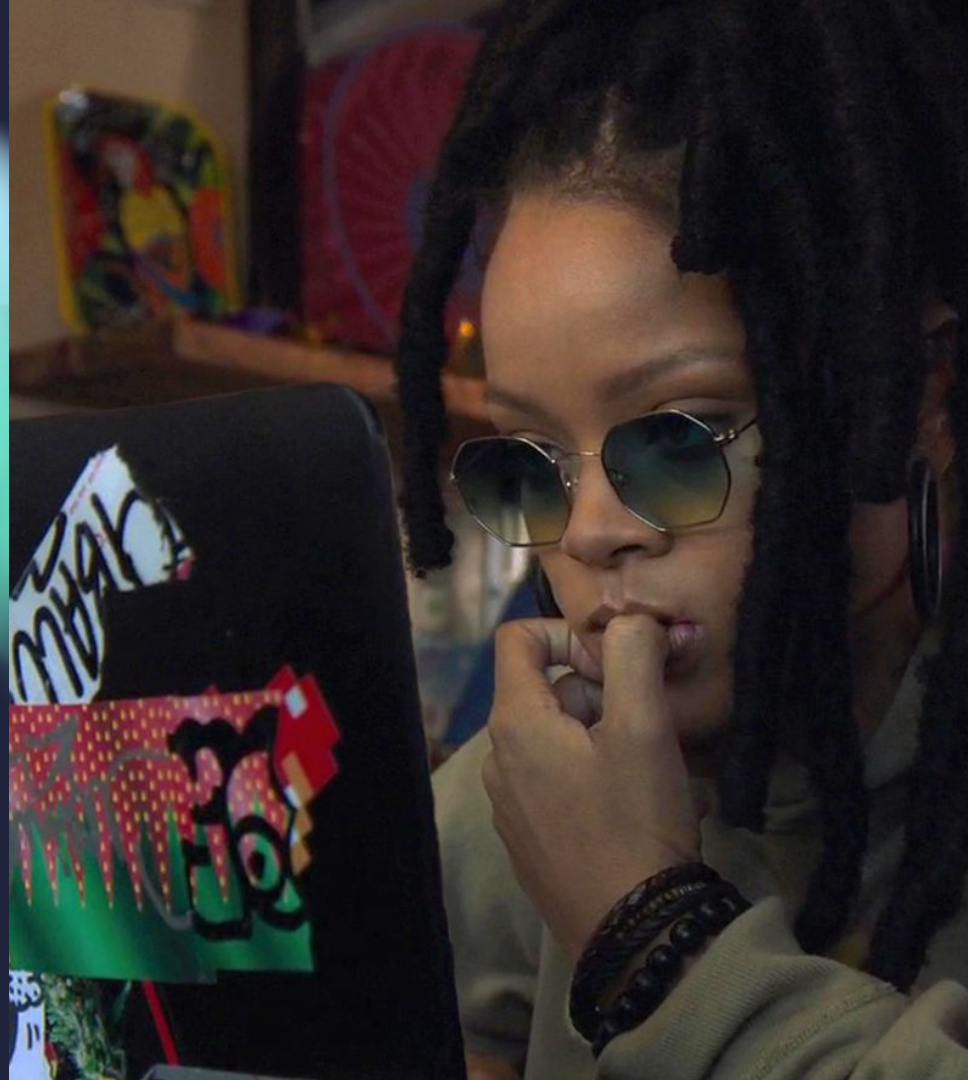
Sign up for Grammarly Engineering Digest!

Wi-Fi: DO-Tagungswelt
Password: DesignOffices



Scaling the Maintainability of Java Codebase at Grammarly

If You Liked It, Then You Shoulda Put a *CI Gate* on It

Yarik Yermilov

Technical Lead @Grammarly

Kyiv 🇺🇦 // Kraków 🇵🇱

https://twitter.com/yermilov17

https://www.grammarly.com/stand-with-ukraine

Grammarly stands with Ukraine

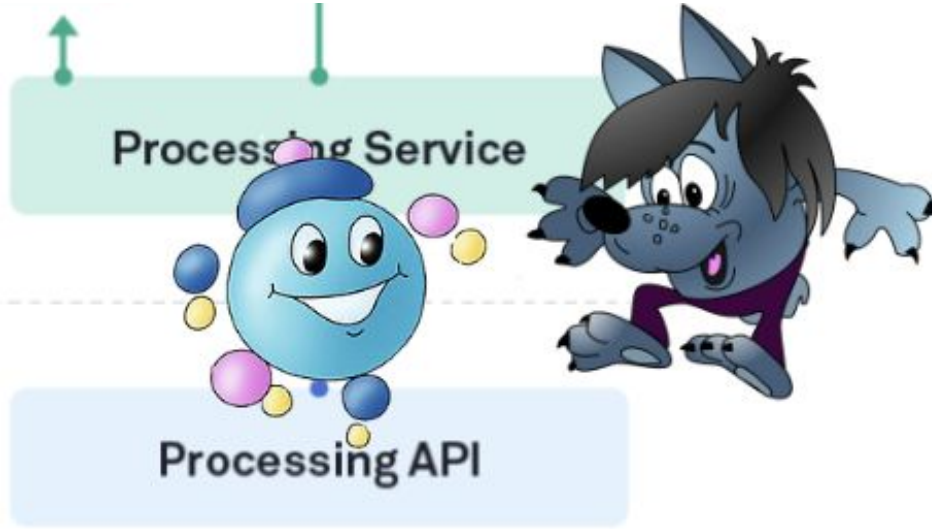
We are a company with a deep connection to Ukraine. Grammarly was founded in Ukraine; our co-founders are from Ukraine, and we have many team members who call Ukraine home. The Russian attacks are horrific atrocities and have caused immense suffering, fear, and heartbreak.

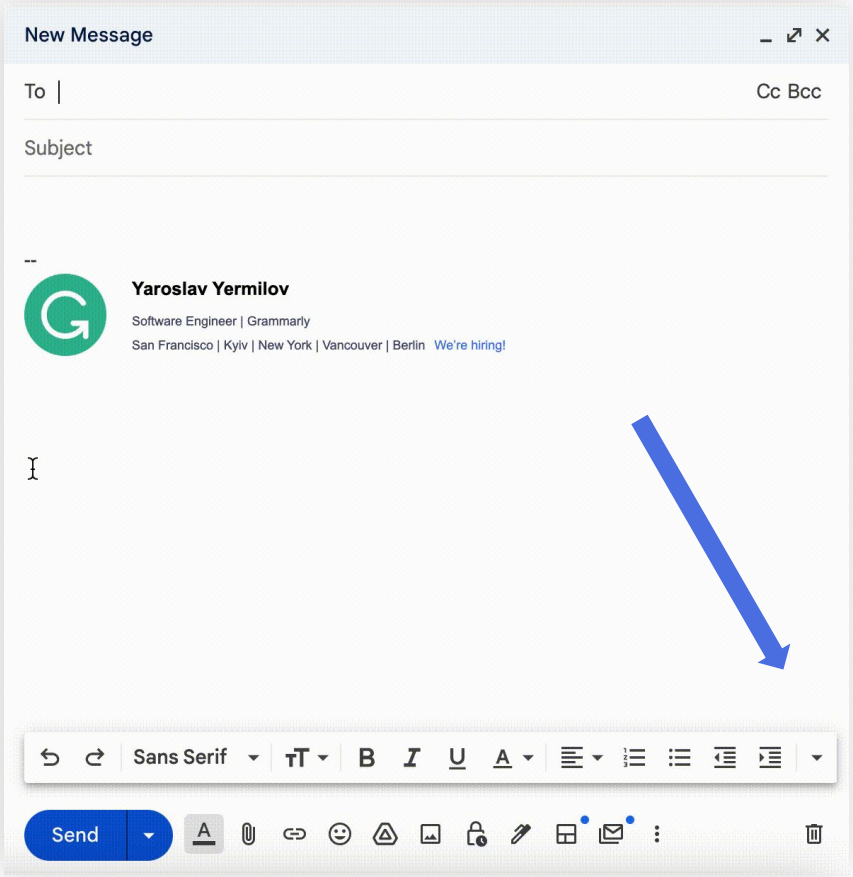
To our colleagues, family, and friends in Ukraine, please know that we stand with you, completely and wholeheartedly. To all of Ukraine, we stand with you and are sending aid. To everyone else, we ask you to declare your support for the people of Ukraine, who continue to resist the unprovoked Russian invasion.

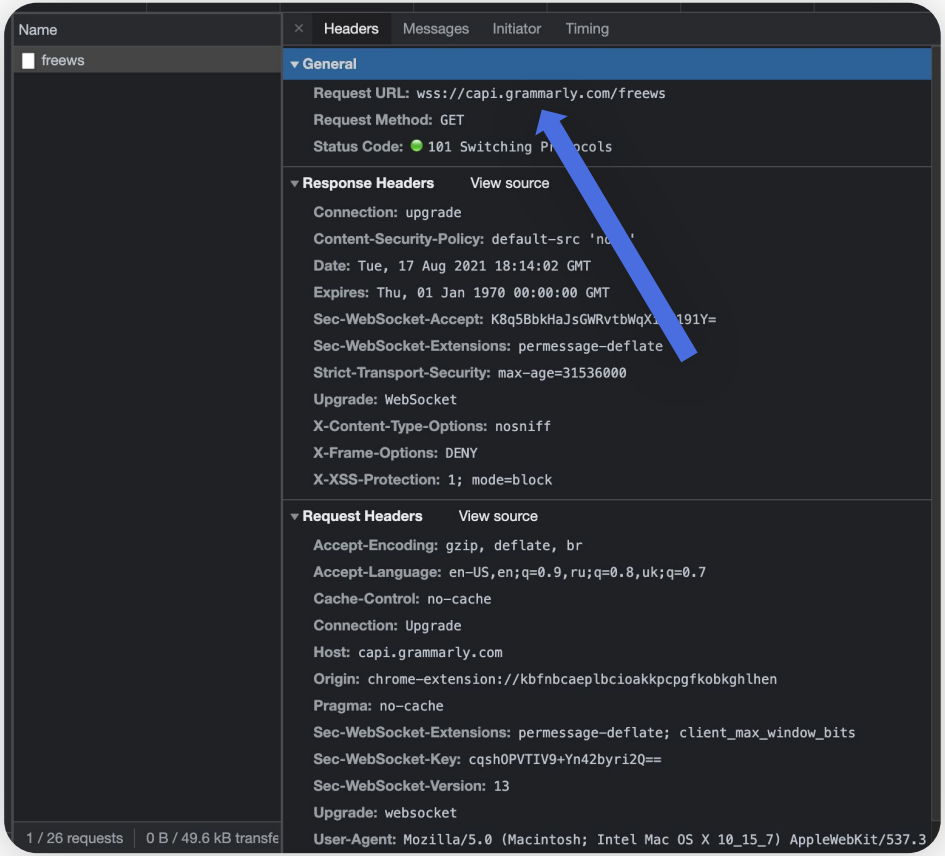
What you can do

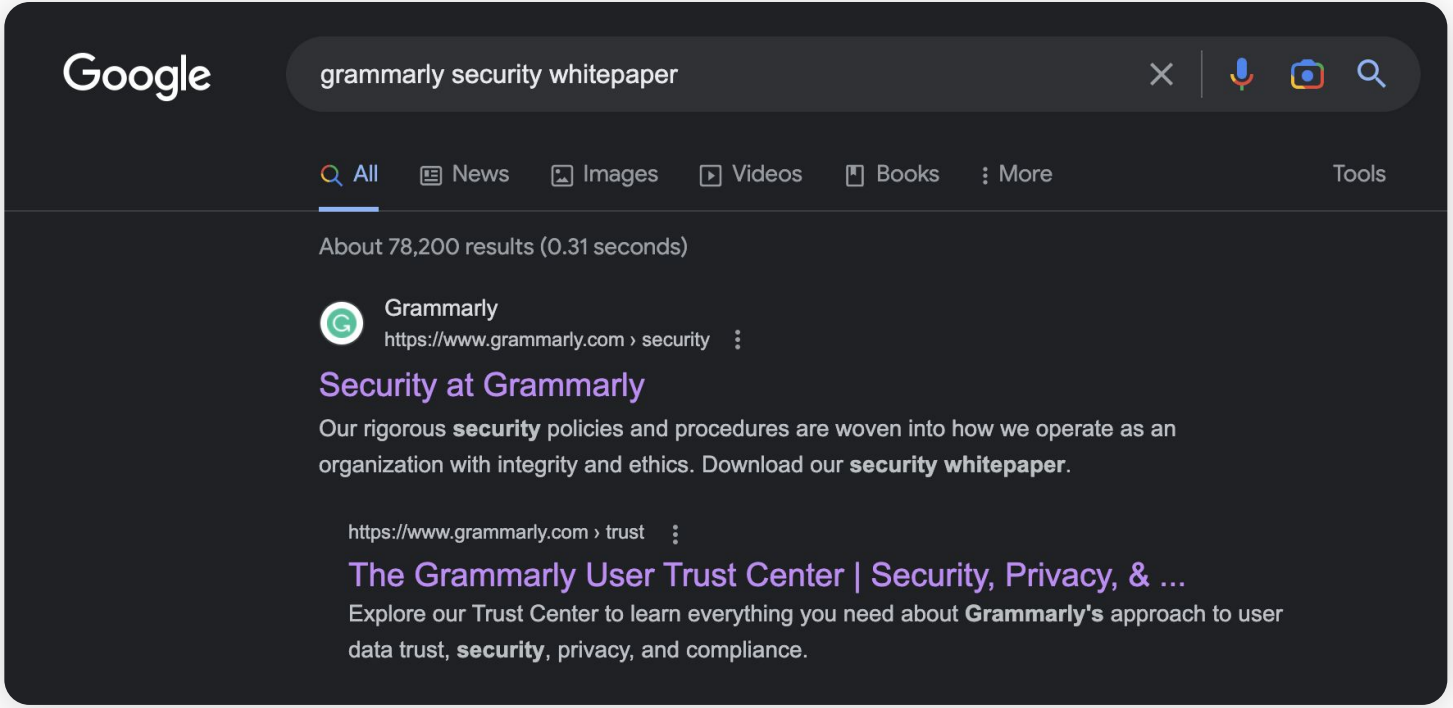
We ask for whatever help you can offer, whether that is by peacefully protesting, speaking to your representatives, or donating money. The demand for war relief is high—and so is the need to sustain hope. Below are links to trustworthy organizations in need of funds to continue supporting and caring for the people of Ukraine.

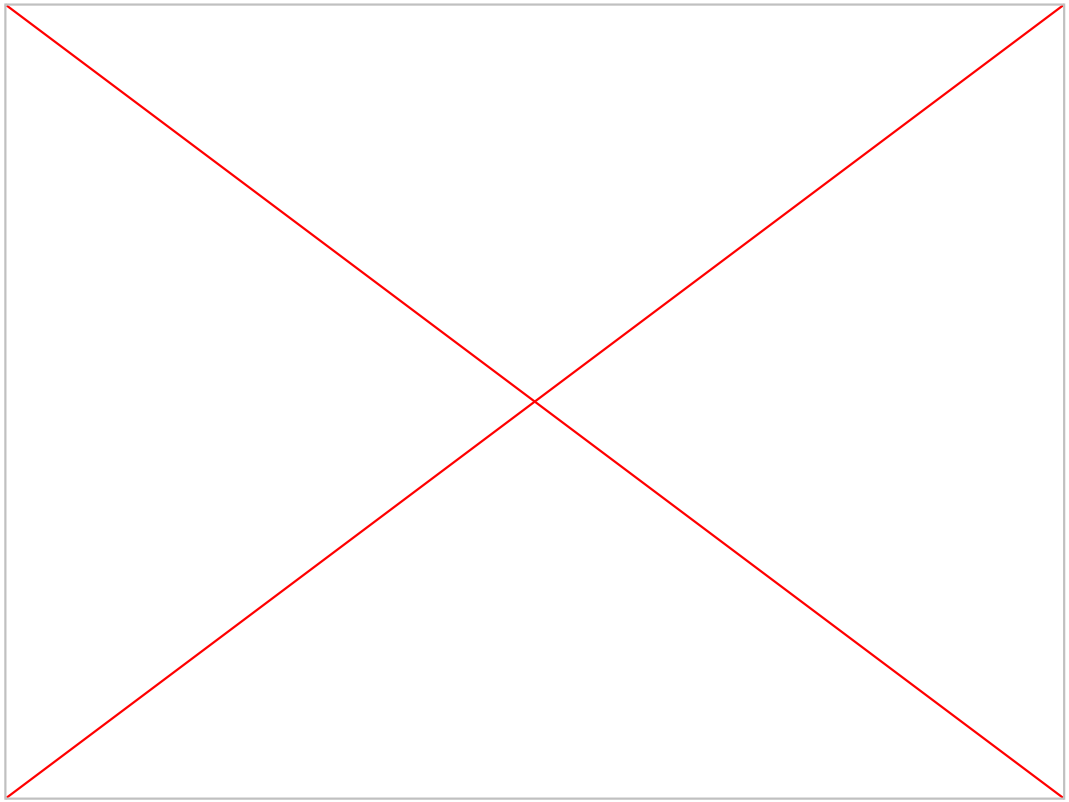
My first day

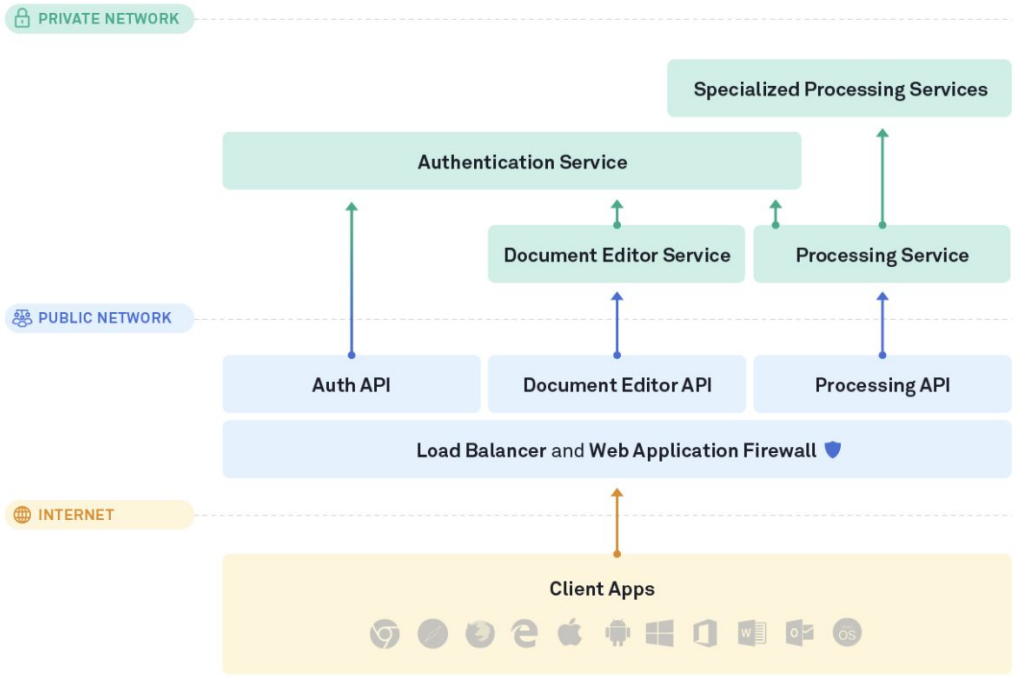


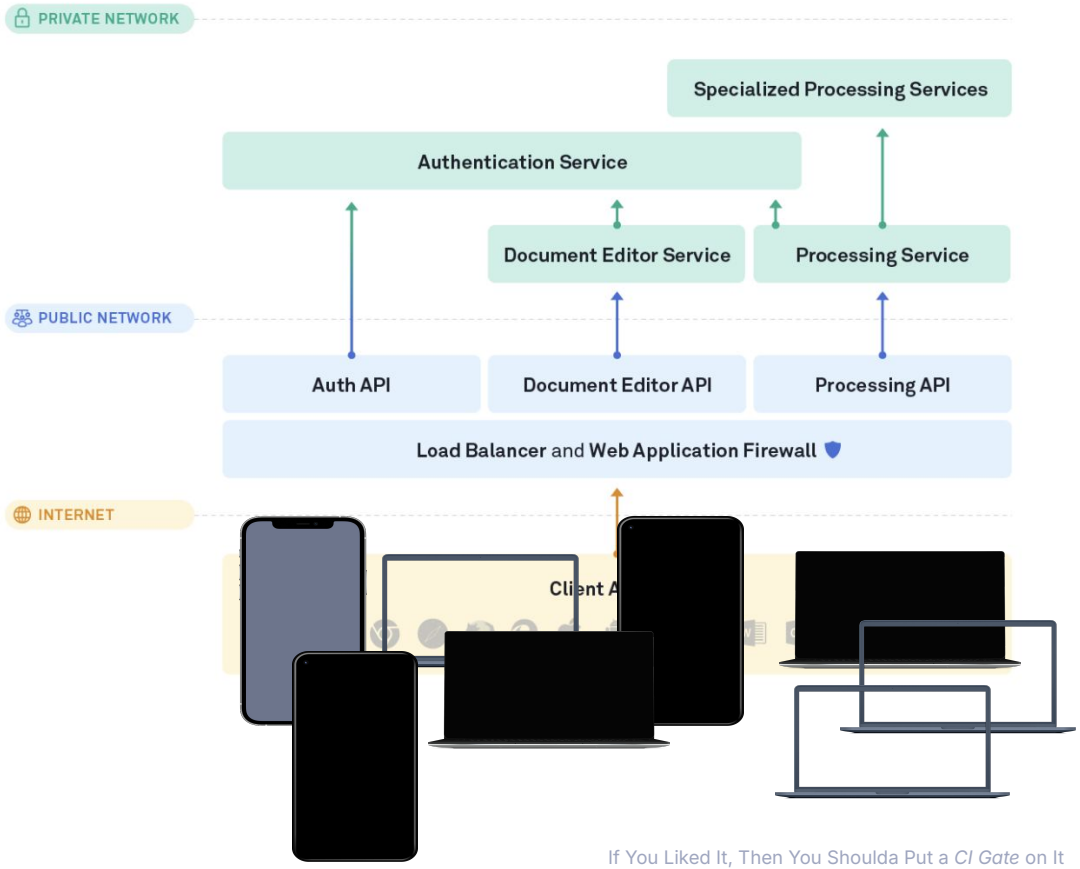

New Message
To
Cc Bcc
Subject
--
Yaroslav Yermilov
Software Engineer | Grammarly
San Francisco | Kyiv | New York | Vancouver | Berlin We're hiring!
Sans Serif
Send

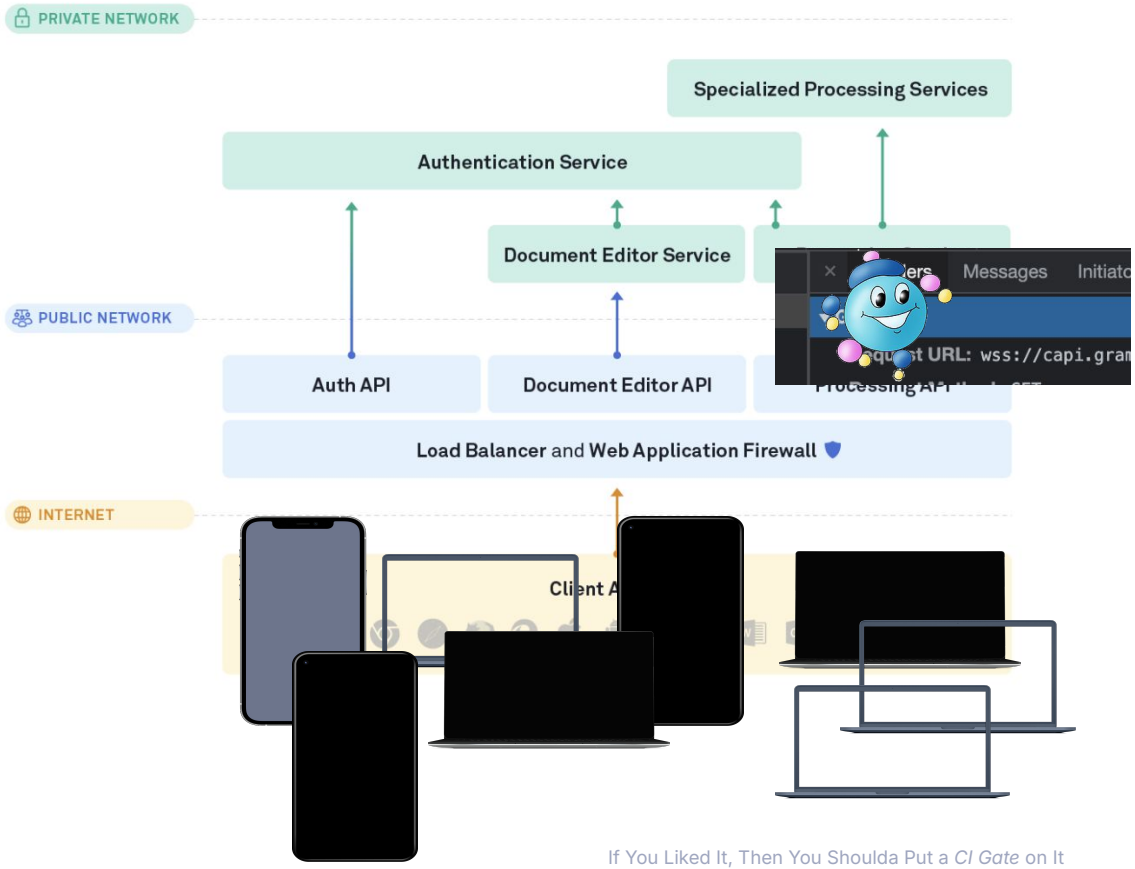

Name
freews
Headers
Messages
Initiator
Timing
General
Request URL: wss://capi.grammarly.com/freews
Request Method: GET
Status Code: 101 Switching Pr ocols
Response Headers
View source
Connection: upgrade
Content-Security-Policy: default-src 'n
Date: Tue, 17 Aug 2021 18:14:02 GMT
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Sec-WebSocket-Accept: K8q5BbkHaJsGWRvtbWqX 191Y=
Sec-WebSocket-Extensions: permessage-deflate
Strict-Transport-Security: max-age=31536000
Upgrade: WebSocket
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
Request Headers
View source
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,ru;q=0.8,uk;q=0.7
Cache-Control: no-cache
Connection: Upgrade
Host: capi.grammarly.com
Origin: chrome-extension://kbfnbcaeplbcioakkpcpgfkobkghlhen
Pragma: no-cache
Sec-WebSocket-Extensions: permessage-deflate; client_max_window_bits
Sec-WebSocket-Key: cqshOPVTIV9+Yn42byri2Q==
Sec-WebSocket-Version: 13
Upgrade: websocket
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.3
1 / 26 requests
0 B / 49.6 kB transfe

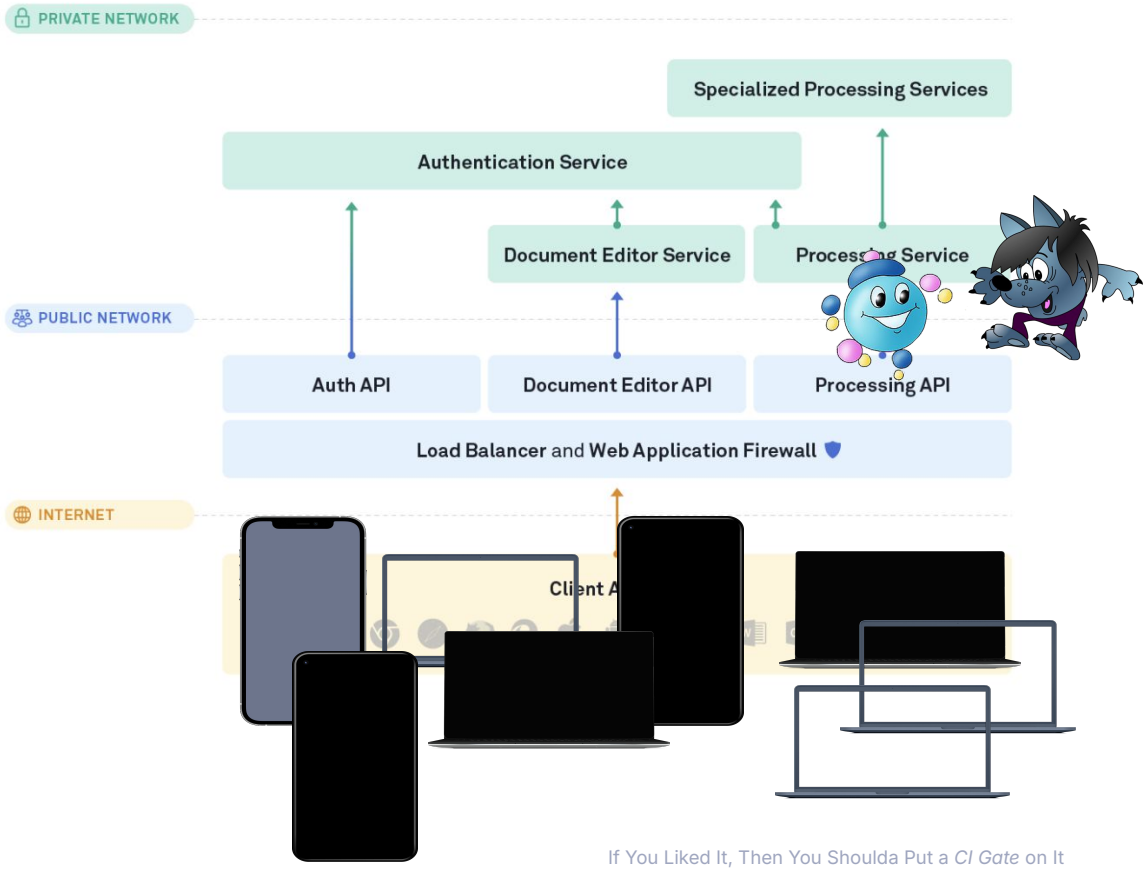

Google
grammarly security whitepaper
All
News
Images
Videos
Books
More
Tools
About 78,200 results (0.31 seconds)
Grammarly
https://www.grammarly.com › security
Security at Grammarly
Our rigorous security policies and procedures are woven into how we operate as an organization with integrity and ethics. Download our security whitepaper.
https://www.grammarly.com › trust
The Grammarly User Trust Center | Security, Privacy, & ...
Explore our Trust Center to learn everything you need about Grammarly's approach to user data trust, security, privacy, and compliance.

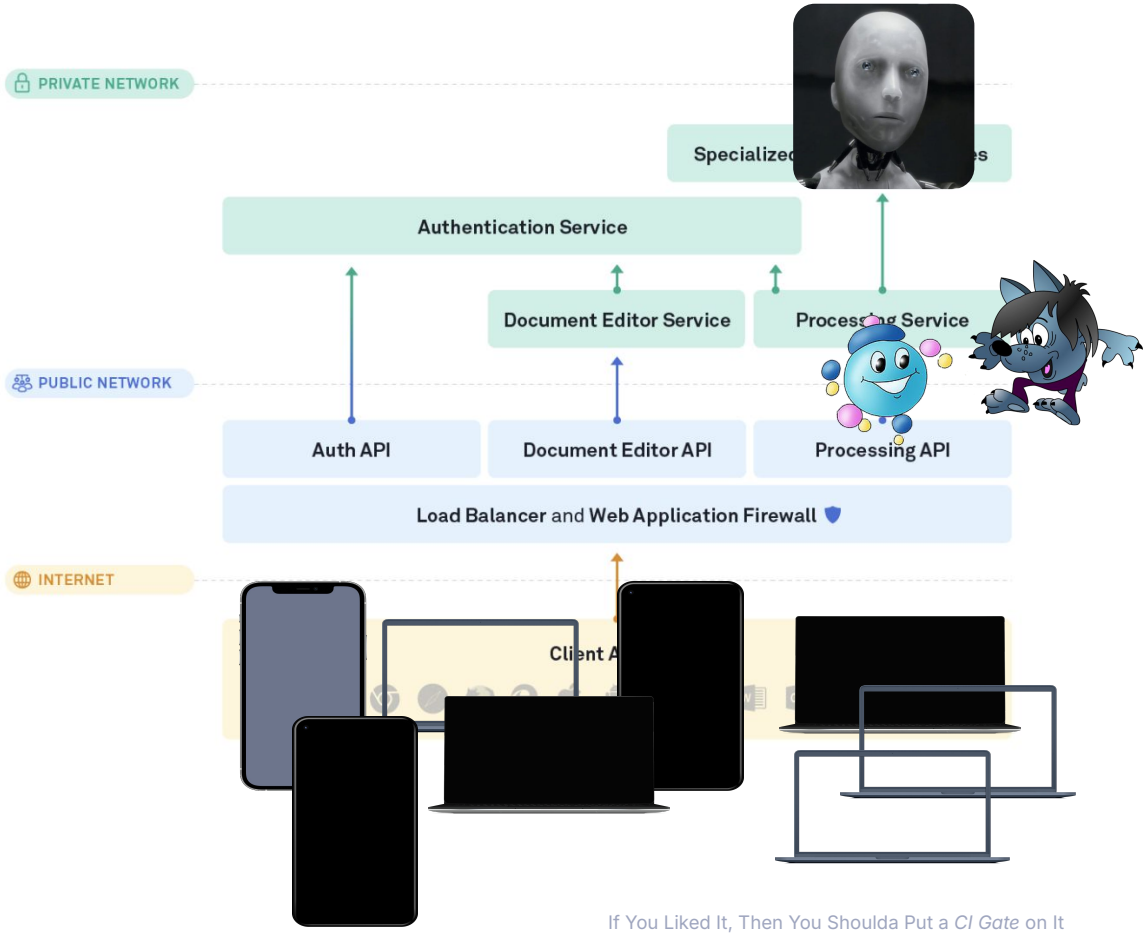


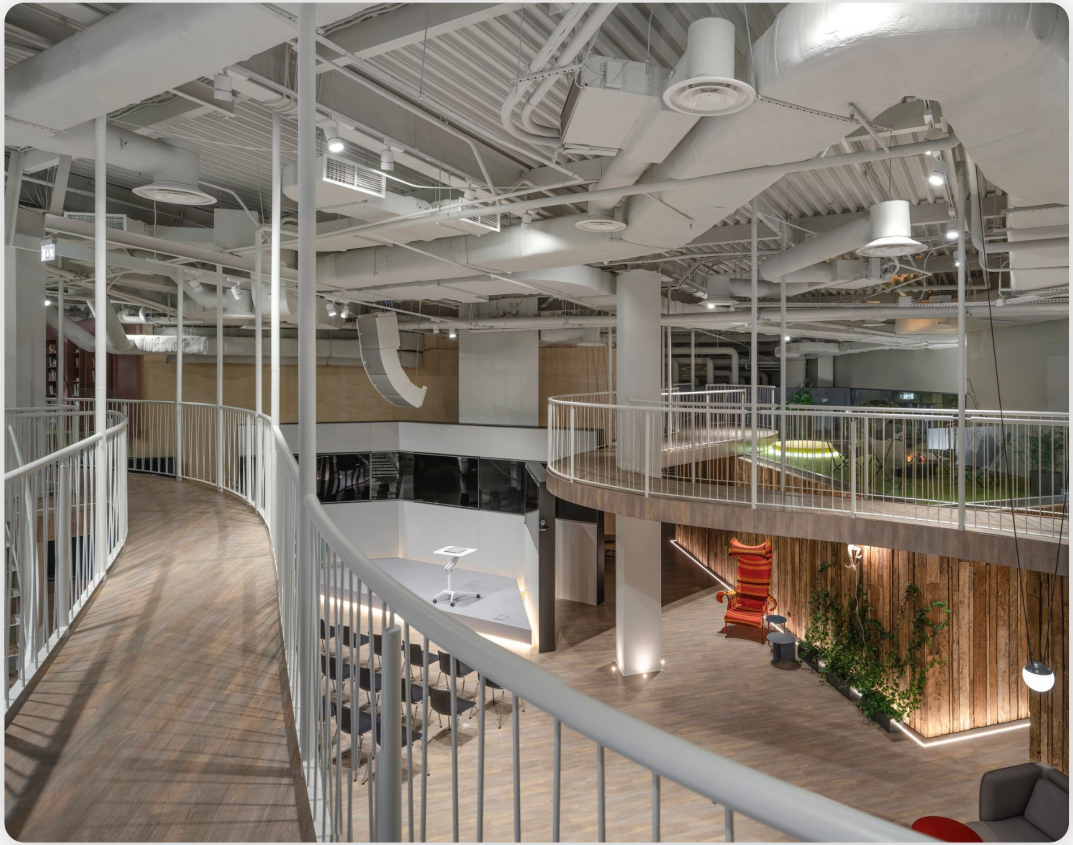

PRIVATE NETWORK
Specialized Processing Services
Authentication Service
Document Editor Service
Processing Service
PUBLIC NETWORK
Auth API
Document Editor API
Processing API
Load Balancer and Web Application Firewall
INTERNET
Client Apps


PRIVATE NETWORK
Specialized Processing Services
Authentication Service
Document Editor Service
Processing Service
PUBLIC NETWORK
Auth API
Document Editor API
Processing API
Load Balancer and Web Application Firewall
INTERNET

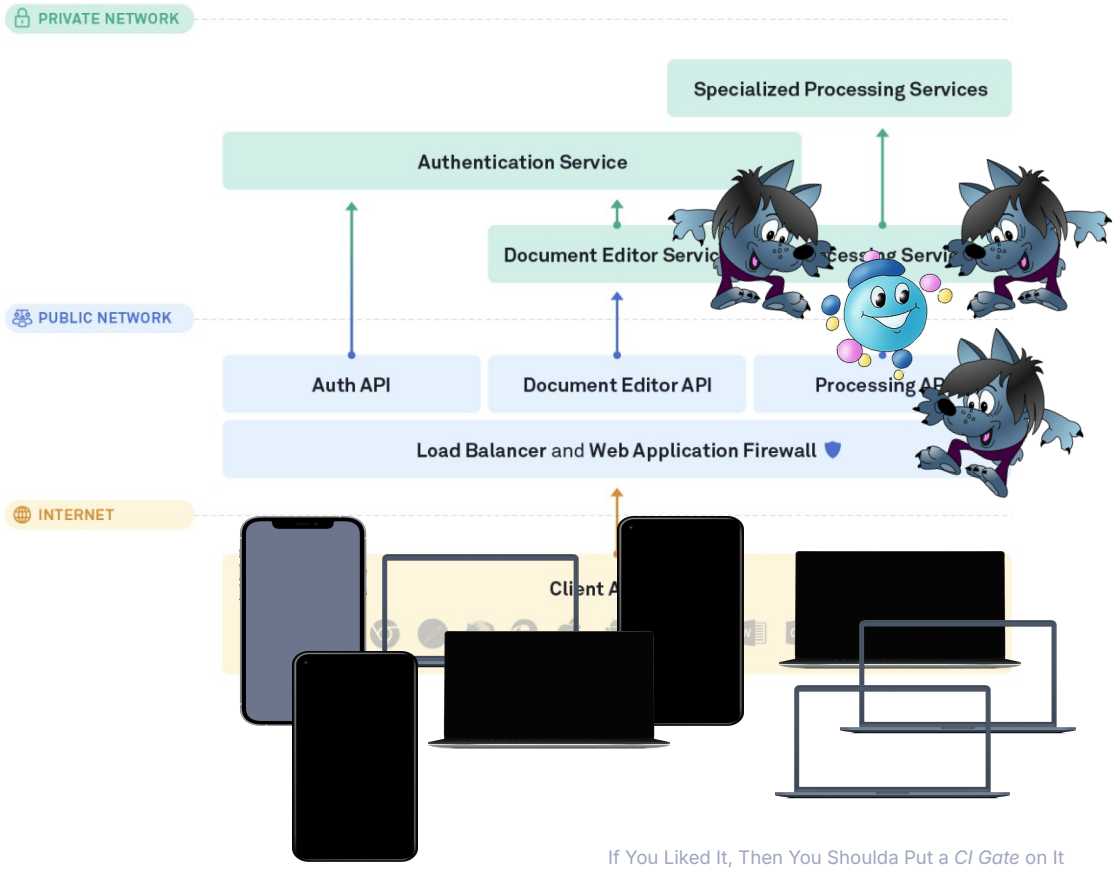

PRIVATE NETWORK
Specialized Processing Services
Authentication Service
Document Editor Service
PUBLIC NETWORK
Messages
Auth API
Document Editor API
Load Balancer and Web Application Firewall
INTERNET


PRIVATE NETWORK
Specialized Processing Services
Authentication Service
Document Editor Service
Processing Service
PUBLIC NETWORK
Auth API
Document Editor API
Processing API
Load Balancer and Web Application Firewall
INTERNET


PRIVATE NETWORK
Specialized
Authentication Service
Document Editor Service
Processing Service
PUBLIC NETWORK
Auth API
Document Editor API
Processing API
Load Balancer and Web Application Firewall
INTERNET
Client A



Team of 3


PRIVATE NETWORK
Specialized Processing Services
Authentication Service
Document Editor Servic
PUBLIC NETWORK
Auth API
Document Editor API
Load Balancer and Web Application Firewall
INTERNET

https://www.youtube.com/watch?v=SsoOG6ZeyUI

Showing 14 changed files

capi-server/src/main/java/grammarly/capi/session/MessageHandler.java

```
...  ...  @@ -7,6 +7,7 @@ import grammarly.capi.SensitiveLogger;
 7    7   import grammarly.capi.SensitiveLogger.NonSensitiveData;
 8    8   import grammarly.capi.backend.auth.UserContext;
 9    9   import grammarly.capi.checker.*;
     10  + import grammarly.capi.configurationapi.clientcontrols.ClientControlsConfigurationManager;
10   11   import grammarly.capi.configurationapi.deactivatedsuggestions.DeactivatedSuggestionsConfigurationManager;
11   12   import grammarly.capi.institution.Snippet;
12   13   import grammarly.capi.knots.Knots;
...  ...  @@ -76,13 +77,7 @@ import org.slf4j.LoggerFactory;
76   77
77   78   import java.lang.reflect.InvocationTargetException;
78   79   import java.lang.reflect.Method;
79       - import java.util.ArrayList;
80       - import java.util.Collections;
81       - import java.util.HashMap;
82       - import java.util.List;
83       - import java.util.Map;
84       - import java.util.Objects;
85       - import java.util.Set;
     80  + import java.util.*;
86   81   import java.util.concurrent.CompletableFuture;
87   82   import java.util.function.Consumer;
```

Showing 12 changed files

capi-server/src/main/java/grammarly/capi/session/MessageHandler.java

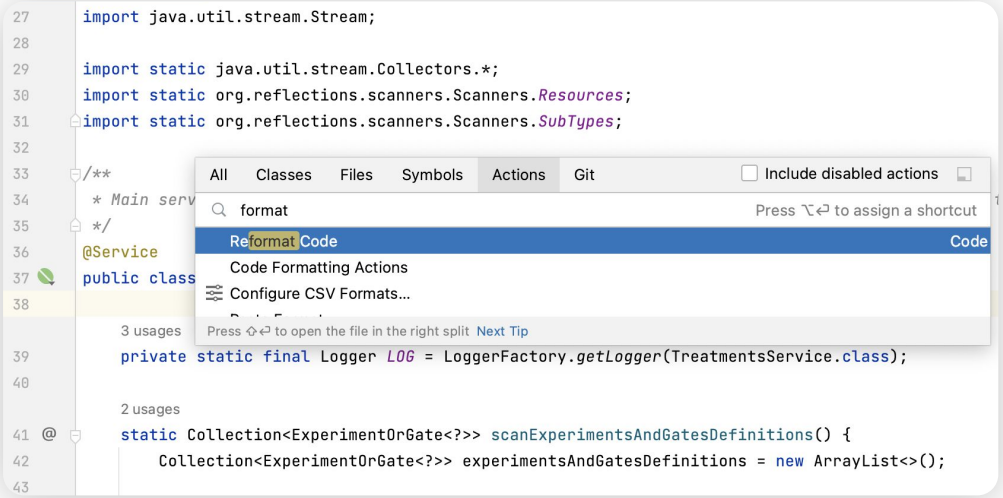
```
...  ...  @@ -58,6 +58,7 @@ import grammarly.quill.operation.RetainOperation;
58   58   import io.micrometer.core.annotation.Timed;
59   59   import io.micrometer.core.instrument.MeterRegistry;
60   60   import io.micrometer.core.instrument.Tags;
     61 + import io.micrometer.core.instrument.Timer;
61   62   import org.apache.commons.lang3.StringUtils;
62   63   import org.apache.commons.lang3.exception.ExceptionUtils;
63   64   import org.apache.commons.lang3.tuple.Pair;
...  ...  @@ -68,7 +69,13 @@ import org.slf4j.LoggerFactory;
68   69   import javax.annotation.Nullable;
69   70   import java.lang.reflect.InvocationTargetException;
70   71   import java.lang.reflect.Method;
71      - import java.util.*;
     72 + import java.util.ArrayList;
     73 + import java.util.Collections;
     74 + import java.util.HashMap;
     75 + import java.util.List;
     76 + import java.util.Map;
     77 + import java.util.Objects;
     78 + import java.util.Set;
72   79   import java.util.concurrent.CompletableFuture;
73   80   import java.util.function.Consumer;
```

Showing 12 changed files

editor-service/src/main/java/grammarly/capi/session/MessageHandler.java

```
42      - import grammarly.capi.healthcheck.ServerState;
    33  + import grammarly.capi.scheduled.ScheduledCheck;
43  34    import grammarly.capi.services.VoxService;
    35  + import grammarly.capi.toggles.experiment.ExperimentsHandler;
    36  + import grammarly.capi.toggles.feature.Feature;
    37  + import grammarly.capi.toggles.feature.FeaturesHandler;
44  38    import grammarly.capi.transport.http.HttpRequestContext;
45  39    import grammarly.capi.transport.ws.EditorCallback;
46  40    import grammarly.capi.transport.ws.OTTransformer;
    41  + import grammarly.capi.usermutes.ScopedUserMuteCategoryInfo;
    42  + import grammarly.capi.usermutes.UserMutesCache;
47  43    import grammarly.capi.usermutes.UserMutesHandler;
48  44    import grammarly.capi.userstats.ChangeTypeDetector;
49  45    import grammarly.capi.userstats.UserProfile;
... ...   @@ -64,21 +60,17 @@ import javax.inject.Named;
64  60    import javax.inject.Provider;
65  61    import java.lang.reflect.InvocationTargetException;
66  62    import java.lang.reflect.Method;
67      - import java.util.ArrayList;
68      - import java.util.HashMap;
69      - import java.util.List;
70      - import java.util.Map;
71      - import java.util.Objects;
    63  + import java.util.*;
72  64    import java.util.concurrent.CompletableFuture;
73  65    import java.util.function.Consumer;
```

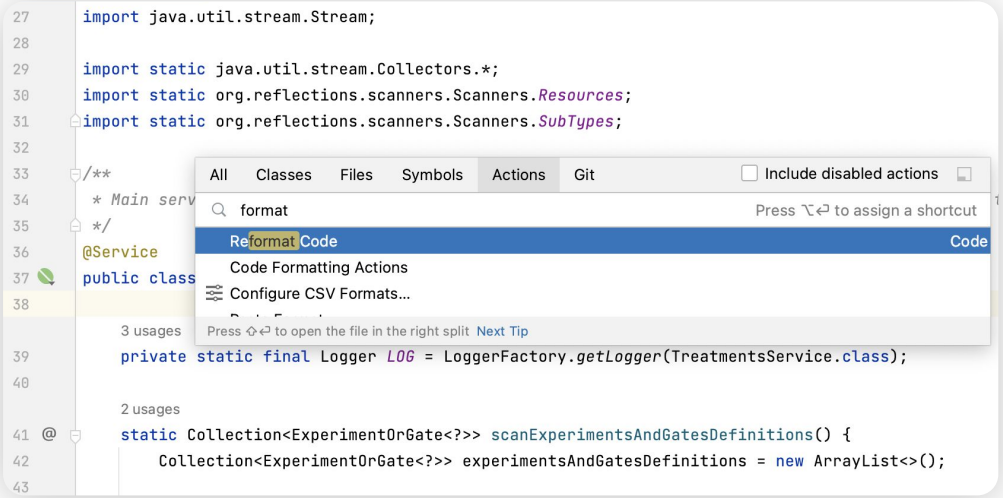
INTELLIJ IDEA



TL;DR

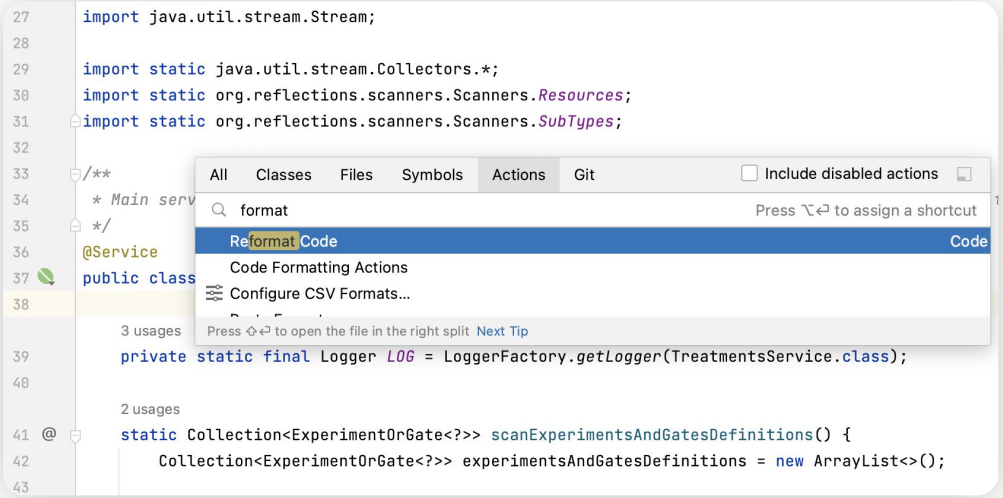
Choice is counter-productive

INTELLIJ IDEA



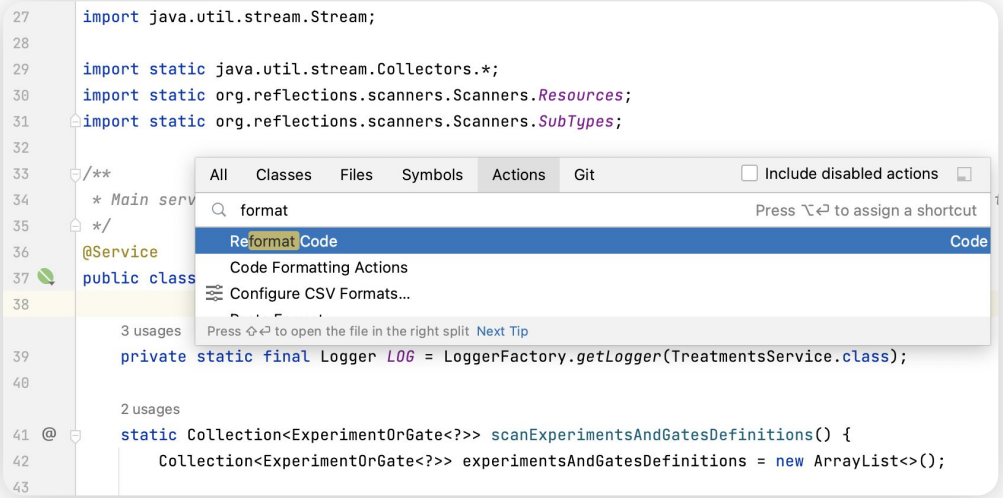
- **Please reformat your code before submitting MR?**

INTELLIJ IDEA



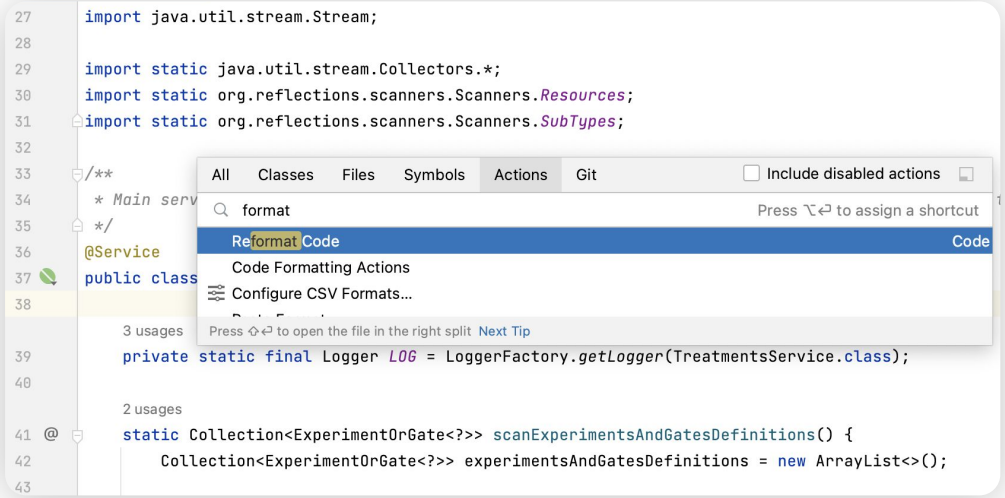
- Please reformat your code before submitting MR?
- Enable “Automatically reformat code on save”?

INTELLIJ IDEA



- Please reformat your code before submitting MR?
- Enable “Automatically reformat code on save”?
- Don’t forget to review code style?

INTELLIJ IDEA



- Please reformat your code before submitting MR?
- Enable “Automatically reformat code on save”?
- Don’t forget to review code style?
- Documentation? Agreement?

TL;DR

Choice is counter-productive

Don’t rely on engineer’s action

DECISION DRIVERS	INTELLIJ IDEA
it is possible to automate code formatting	❓

DECISION DRIVERS	INTELLIJ IDEA
it is possible to automate code formatting	❓

Format files from the command line

Last modified: 02 May 2022

IntelliJ IDEA can format your code according to the configured code style settings. You can also apply your code style formatting to the specified files from the command line.

The command-line formatter launches an instance of IntelliJ IDEA in the background and applies the formatting. It will not work if another instance of IntelliJ IDEA is already running. In this case, you can perform code style formatting from the running instance. Use the command-line formatter for automated regular maintenance of a large codebase with many contributors to ensure a consistent coding style.

To be able to format files, install and enable plugins with support for the corresponding file types in IntelliJ IDEA (for example, the **Shell Script** plugin to format shell script files).

Windows macOS Linux

You can find the script for running IntelliJ IDEA in the installation directory under **`bin`**. To use this script as the command-line launcher, add it to your system `PATH` as described in Command-line interface.

Syntax

```
idea.sh format [<options>] <path ...>
```

DECISION DRIVERS	INTELLIJ IDEA
it is possible to automate code formatting	❓

Format files from

Last modified: 02 May 2022

IntelliJ IDEA can format your code according to
also apply your code style formatting to the spe

The command-line formatter launches an instar
applies the formatting. It will not work if anothe
this case, you can perform code style formattin
command-line formatter for automated regular
contributors to ensure a consistent coding style

To be able to format files, install and enable plug
types in IntelliJ IDEA (for example, the **Shell Scri**

Windows macOS Linux

You can find the script for running IntelliJ IDEA i
this script as the command-line launcher, add it
Command-line interface.

Syntax

```
idea.sh format [<options>] <path ..
```

```
FROM adoptopenjdk/openjdk8

LABEL maintainer "Viktor Adam <rycus86@gmail.com>"

ARG IDEA_VERSION=2021.1
ARG IDEA_BUILD=2021.1.3

RUN  \
  apt-get update && apt-get install --no-install-recommends -y \
  gcc git openssh-client less \
  libxtst-dev libxext-dev libxrender-dev libfreetype6-dev \
  libfontconfig1 libgtk2.0-0 libxslt1.1 libxxf86vm1 \
  && rm -rf /var/lib/apt/lists/* \
  && useradd -ms /bin/bash developer

ARG idea_source=https://download.jetbrains.com/idea/ideaIC-${IDEA_BUILD}.tar.gz
ARG idea_local_dir=.IdeaIC${IDEA_VERSION}

WORKDIR /opt/idea

RUN curl -fsSL $idea_source -o /opt/idea/installer.tgz \
  && tar --strip-components=1 -xzf installer.tgz \
  && rm installer.tgz

USER developer
ENV HOME /home/developer

RUN mkdir /home/developer/.Idea \
  && ln -sf /home/developer/.Idea /home/developer/$idea_local_dir

CMD [ "/opt/idea/bin/idea.sh" ]
```

DECISION DRIVERS	INTELLIJ IDEA
it is possible to automate code formatting	😬

Format files from

Last modified: 02 May 2022

IntelliJ IDEA can format your code according to
also apply your code style formatting to the spe

The command-line formatter launches an instar
applies the formatting. It will not work if anothe
this case, you can perform code style formattin
command-line formatter for automated regular
contributors to ensure a consistent coding style

To be able to format files, install and enable plug
types in IntelliJ IDEA (for example, the **Shell Scri**

Windows macOS Linux

You can find the script for running IntelliJ IDEA i
this script as the command-line launcher, add it
Command-line interface.

Syntax

```
idea.sh format [<options>] <path ..
```

```
FROM adoptopenjdk/openjdk8

LABEL maintainer "Viktor Adam <rycus86@gmail.com>"

ARG IDEA_VERSION=2021.1
ARG IDEA_BUILD=2021.1.3

RUN  \
  apt-get update && apt-get install --no-install-recommends -y \
  gcc git openssh-client less \
  libxtst-dev libxext-dev libxrender-dev libfreetype6-dev \
  libfontconfig1 libgtk2.0-0 libxslt1.1 libxxf86vm1 \
  && rm -rf /var/lib/apt/lists/* \
  && useradd -ms /bin/bash developer

ARG idea_source=https://download.jetbrains.com/idea/ideaIC-${IDEA_BUILD}.tar.gz
ARG idea_local_dir=.IdeaIC${IDEA_VERSION}

WORKDIR /opt/idea

RUN curl -fsSL $idea_source -o /opt/idea/installer.tgz \
  && tar --strip-components=1 -xzf installer.tgz \
  && rm installer.tgz

USER developer
ENV HOME /home/developer

RUN mkdir /home/developer/.Idea \
  && ln -sf /home/developer/.Idea /home/developer/$idea_local_dir

CMD [ "/opt/idea/bin/idea.sh" ]
```

DECISION DRIVERS	INTELLIJ IDEA	SPOTLESS
it is possible to automate code formatting	😬	👍

README.md

Spotless: Keep your code spotless

gradle plugin v6.18.0 | maven plugin v2.36.0 | sbt plugin 0.1.3

Spotless can format <antlr | c | c# | c++ | css | flow | graphql | groovy | html | java | javascript | json | jsx | kotlin | less | license headers | markdown | objective-c | protobuf | python | scala | scss | sql | typeScript | vue | yaml | anything> using <gradle | maven | sbt | anything>.

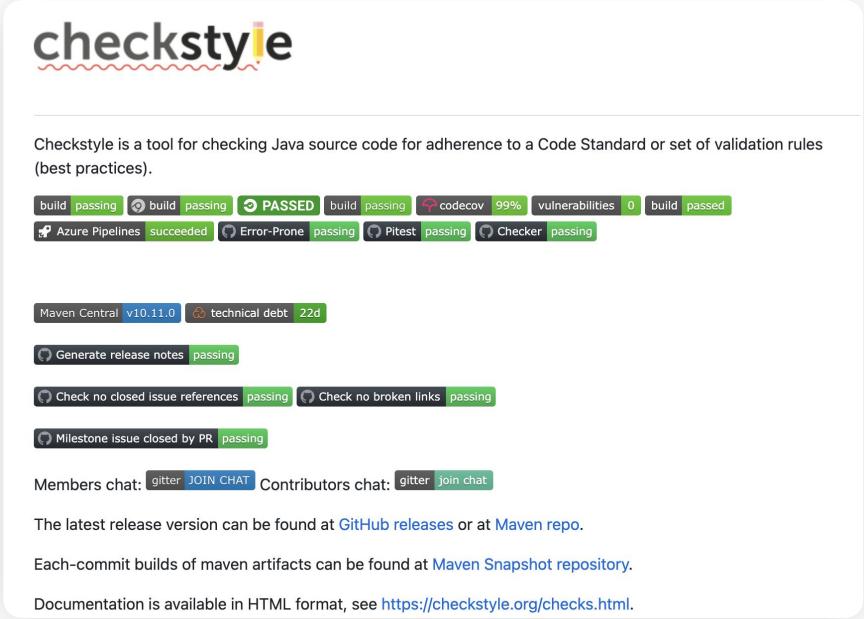
You probably want one of the links below:

Spotless for Gradle (with integrations for VS Code and IntelliJ)

```
user@machine repo % ./gradlew build
:spotlessJavaCheck FAILED
  The following files had format violations:
  src\main\java\com\diffplug\gradle\spotless\FormatExtension.java
    -\t\t····if·(targets.length·==·0)·{
    +\t\tif·(targets.length·==·0)·{
  Run './gradlew spotlessApply' to fix these violations.
user@machine repo % ./gradlew spotlessApply
:spotlessApply
BUILD SUCCESSFUL
user@machine repo % ./gradlew build
BUILD SUCCESSFUL
```

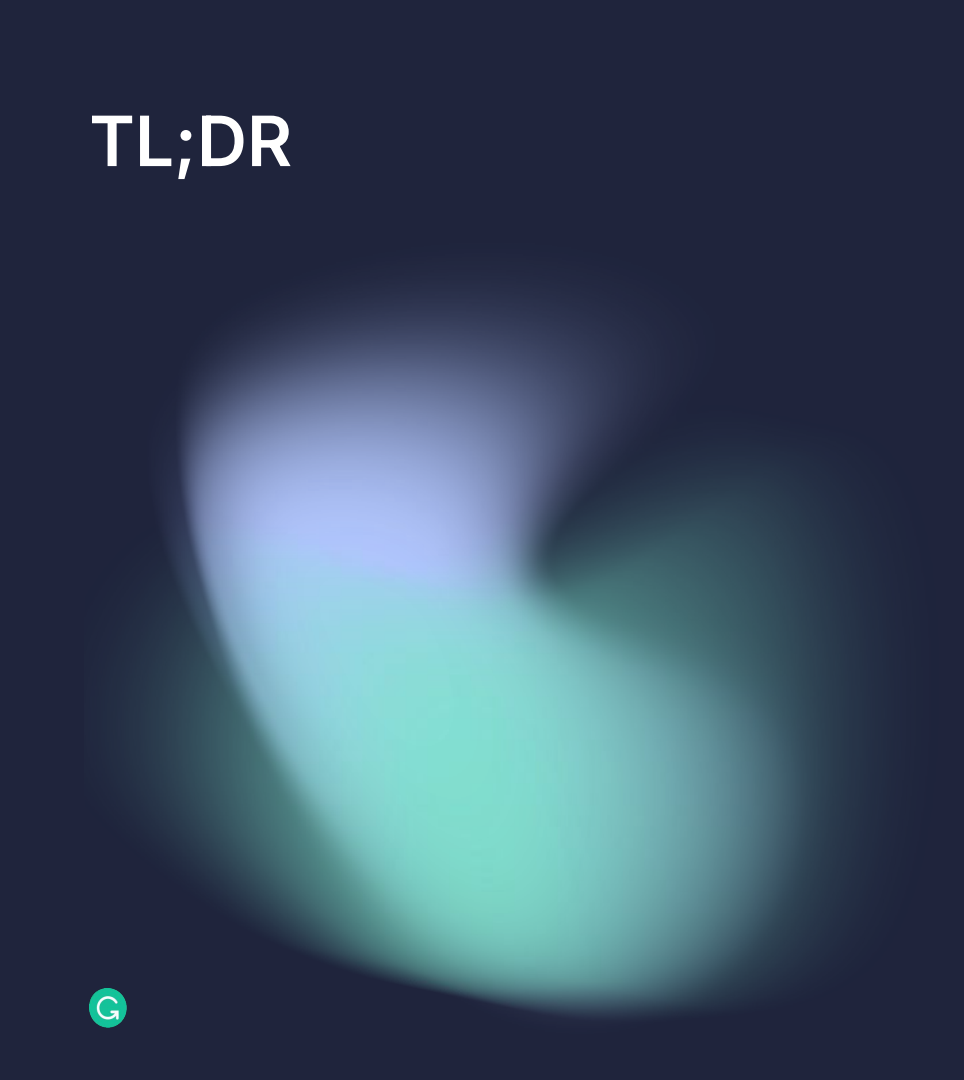
github.com/diffplug/spotless

DECISION DRIVERS	INTELLIJ IDEA	SPOTLESS	CHECKSTYLE
it is possible to automate code formatting	😬	👍	👎



github.com/checkstyle/checkstyle

TL;DR



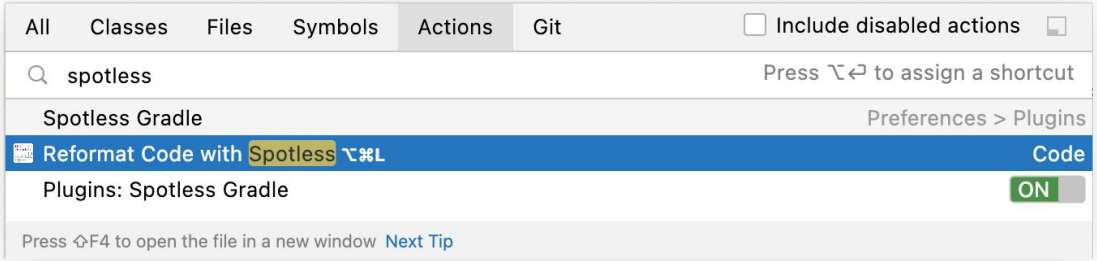
Choice is counter-productive

Don’t rely on engineer’s action

Think a lot about development experience

DECISION DRIVERS	INTELLIJ IDEA	SPOTLESS	CHECKSTYLE
it is possible to automate code formatting	😬	👍	👎
convenient development experience	⭐		

DECISION DRIVERS	INTELLIJ IDEA	SPOTLESS	CHECKSTYLE
it is possible to automate code formatting	😬	👍	👎
convenient development experience	⭐	👍	👍



DECISION DRIVERS	INTELLIJ IDEA	SPOTLESS	CHECKSTYLE
it is possible to automate code formatting	😬	👍	👎
convenient development experience	⭐	👍	👍
migration should be smooth and manageable	👎	👍	👍

DECISION DRIVERS	INTELLIJ IDEA	SPOTLESS	CHECKSTYLE
it is possible to automate best practice	😬	👍	👎
convenient development experience	⭐	👍	👍
introducing best practice for the first time	👎	👍	👍

DECISION DRIVERS	INTELLIJ IDEA	SPOTLESS	CHECKSTYLE
it is possible to automate code formatting	😬	👍	👎
convenient development experience	⭐	👍	👍
migration should be smooth and manageable	👎	👍	👍
should support different file formats	⭐	⭐	👎

DECISION DRIVERS	INTELLIJ IDEA	SPOTLESS	CHECKSTYLE
it is possible to automate code formatting	😬	👍	👎
convenient development experience	⭐	👍	👍
migration should be smooth and manageable	👎	👍	👍
should support different file formats	⭐	⭐	👎
code style should be similar to the informal one we already use, including the ability to fine-tune style to our needs	⭐	👍	👍

DECISION DRIVERS	SPOTLESS
it is possible to automate code formatting	👍
convenient development experience	👍
migration should be smooth and manageable	👍
should support different file formats	⭐
code style should be similar to the informal one we already use, including the ability to fine-tune style to our needs	👍

TL;DR

Choice is counter-productive

Don’t rely on engineer’s action

Think a lot about development experience

Be explicit about decisions

DECISION DRIVERS	SPOTLESS
it is possible to automate code formatting	👍
convenient development experience	👍
migration should be smooth and manageable	👍
should support different file formats	⭐
code style should be similar to the informal one we already use, including the ability to fine-tune style to our needs	👍

google-java-format

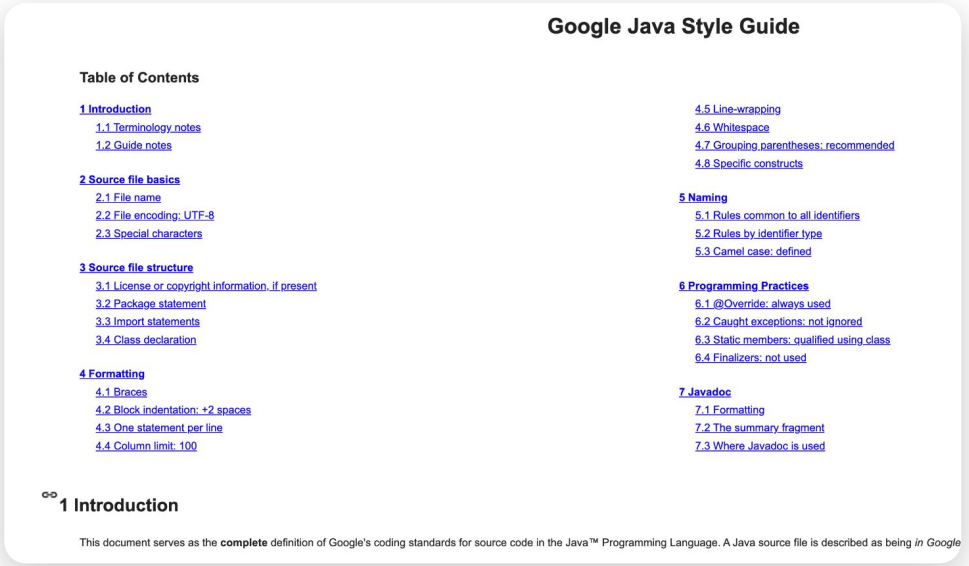
homepage. changelog.

```
spotless {
  java {
    googleJavaFormat()
    // optional: you can specify a specific version (>= 1.8) and/or switch to AOSP style
    //   and/or reflow long strings
    //   and/or use custom group artifact (you probably don't need this)
    googleJavaFormat('1.8').aosp().reflowLongStrings().groupArtifact('com.google.googlejavaformat:google-java-format')
```

palantir-java-format

homepage. changelog.

```
spotless {
  java {
    palantirJavaFormat()
    // optional: you can specify a specific version and/or switch to AOSP/GOOGLE style
    palantirJavaFormat('2.9.0').style("GOOGLE")
```



Google Java Style Guide

Table of Contents

1 Introduction

This document serves as the **complete** definition of Google's coding standards for source code in the Java™ Programming Language. A Java source file is described as being *in Google*

https://google.github.io/styleguide/javaguide.html

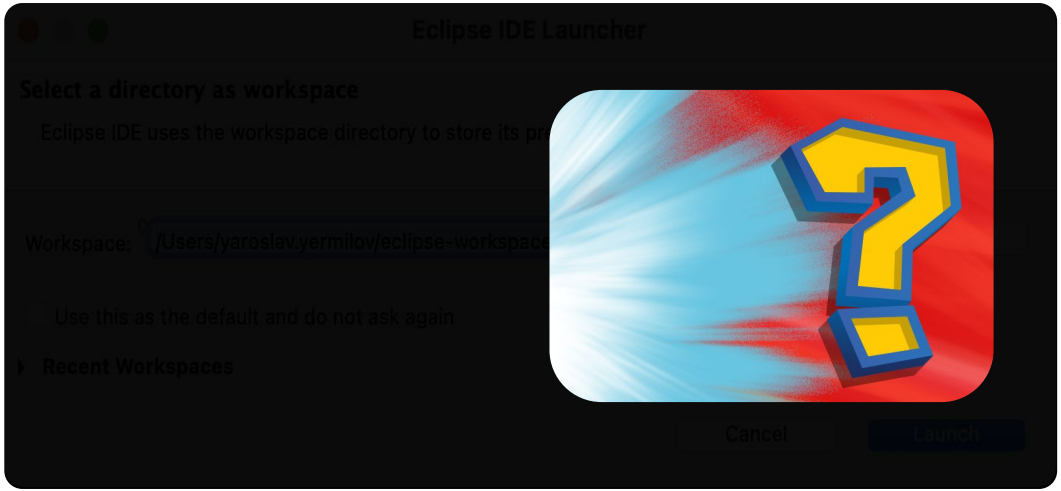
Google Java Style Guide

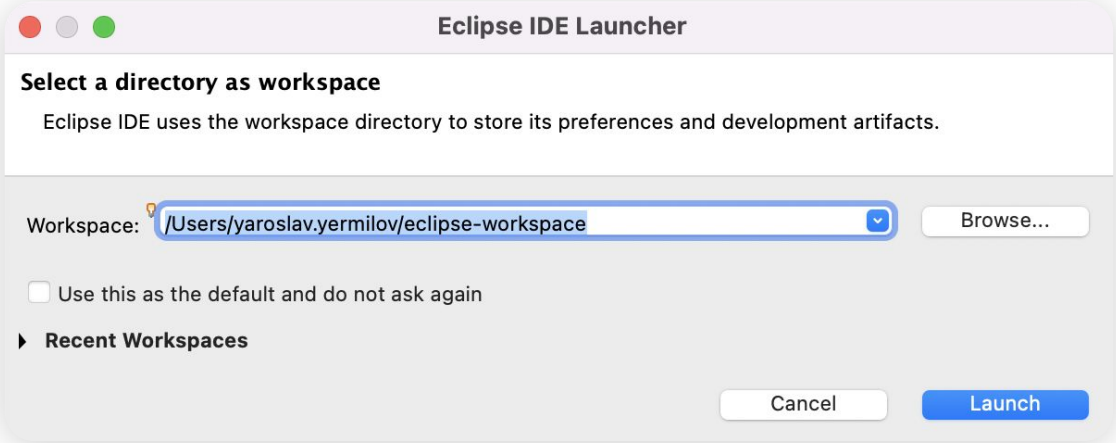
Table of Contents

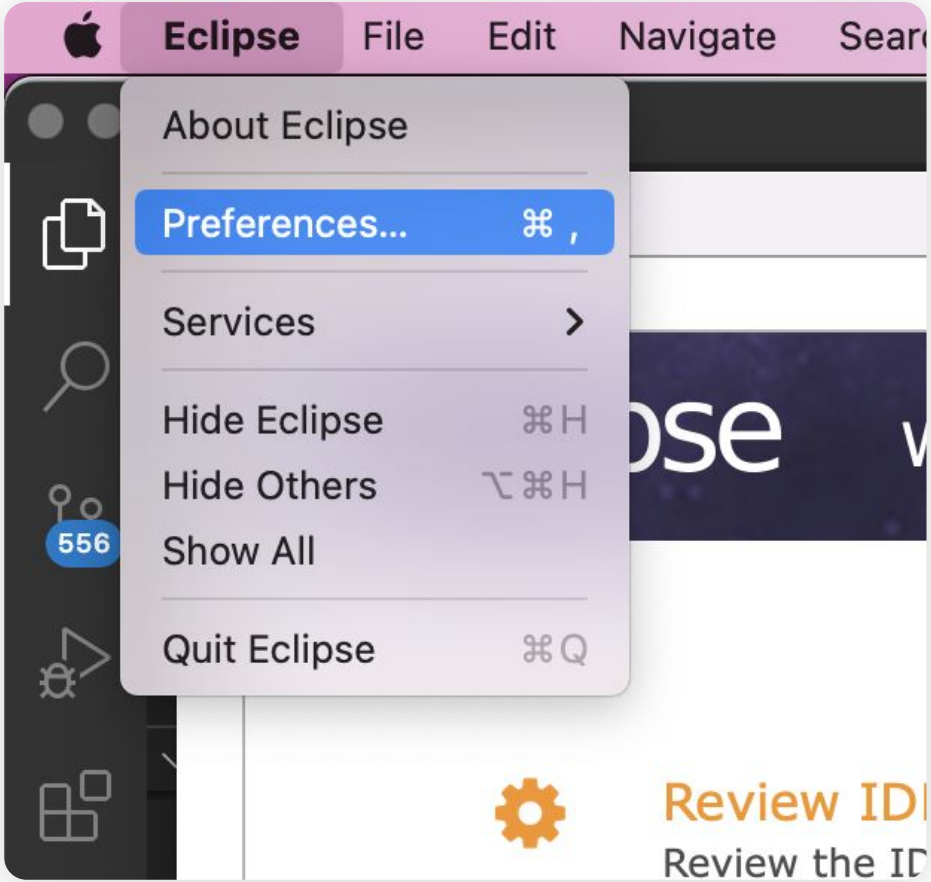
1 Introduction

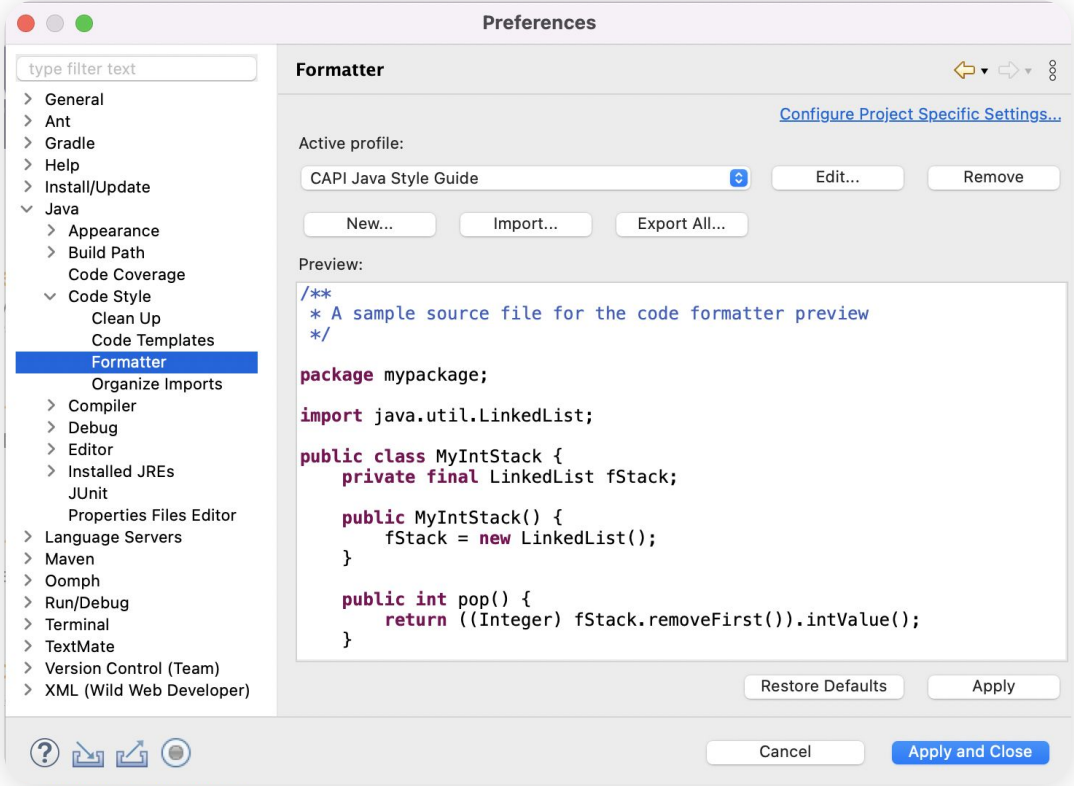
This document serves as the **complete** definition of Google's coding standards for source code in the Java™ Programming Language. A Java source file is described as being *in Google*

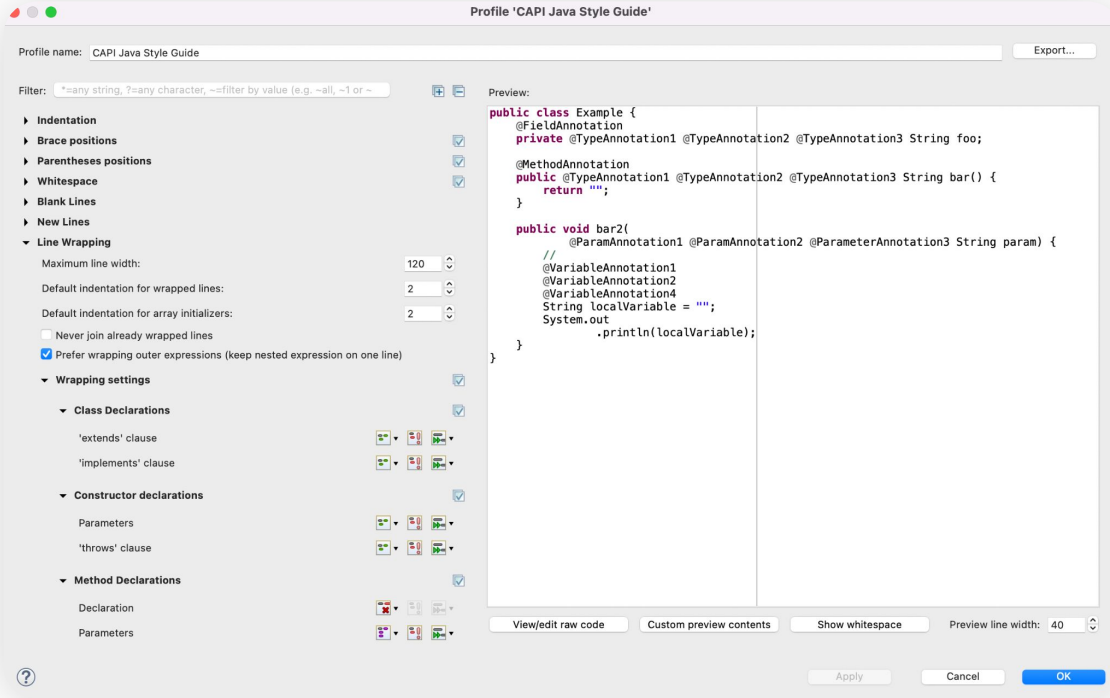
https://google.github.io/styleguide/javaguide.html


Eclipse IDE Launcher
Select a directory as workspace
Eclipse IDE uses the workspace directory to store its pr
Workspace:
/Users/yaroslav.yermilov/eclipse-workspace
Use this as the default and do not ask again
Recent Workspaces
Cancel
Launch


Eclipse IDE Launcher
Select a directory as workspace
Eclipse IDE uses the workspace directory to store its preferences and development artifacts.
Workspace:
/Users/yaroslav.yermilov/eclipse-workspace
Browse...
Use this as the default and do not ask again
Recent Workspaces
Cancel
Launch


Eclipse
File
Edit
Navigate
About Eclipse
Preferences... ⌘ ,
Services
Hide Eclipse ⌘H
Hide Others ⌥⌘H
Show All
Quit Eclipse ⌘Q
556
Review ID
Review the ID


Preferences
type filter text
General
Ant
Gradle
Help
Install/Update
Java
Appearance
Build Path
Code Coverage
Code Style
Clean Up
Code Templates
Formatter
Organize Imports
Compiler
Debug
Editor
Installed JREs
JUnit
Properties Files Editor
Language Servers
Maven
Oomph
Run/Debug
Terminal
TextMate
Version Control (Team)
XML (Wild Web Developer)
Formatter
Configure Project Specific Settings...
Active profile:
CAPI Java Style Guide
Edit...
Remove
New...
Import...
Export All...
Preview:
/**
* A sample source file for the code formatter preview
*/
package mypackage;
import java.util.LinkedList;
public class MyIntStack {
private final LinkedList fStack;
public MyIntStack() {
fStack = new LinkedList();
}
public int pop() {
return ((Integer) fStack.removeFirst()).intValue();
}
Restore Defaults
Apply
Cancel
Apply and Close


Profile 'CAPI Java Style Guide'
Profile name: CAPI Java Style Guide
Export...
Filter: *=any string, ?=any character, ~=filter by value (e.g. ~all, ~1 or ~
Indentation
Brace positions
Parentheses positions
Whitespace
Blank Lines
New Lines
Line Wrapping
Maximum line width: 120
Default indentation for wrapped lines: 2
Default indentation for array initializers: 2
Never join already wrapped lines
Prefer wrapping outer expressions (keep nested expression on one line)
Wrapping settings
Class Declarations
'extends' clause
'implements' clause
Constructor declarations
Parameters
'throws' clause
Method Declarations
Declaration
Parameters
Preview:
public class Example {
@FieldAnnotation
private @TypeAnnotation1 @TypeAnnotation2 @TypeAnnotation3 String foo;
@MethodAnnotation
public @TypeAnnotation1 @TypeAnnotation2 @TypeAnnotation3 String bar() {
return "";
}
public void bar2(
@ParamAnnotation1 @ParamAnnotation2 @ParameterAnnotation3 String param) {
//
@VariableAnnotation1
@VariableAnnotation2
@VariableAnnotation4
String localVariable = "";
System.out
.println(localVariable);
}
}
View/edit raw code
Custom preview contents
Show whitespace
Preview line width: 40
Apply
Cancel
OK

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<profiles version="22">
        <profile kind="CodeFormatterProfile"
                name="CAPI Java Style Guide" version="22">
                <setting
                        id="org.eclipse.jdt.core.formatter.insert_space_after_ellipsis"
                        value="insert" />
                <setting
                        id="org.eclipse.jdt.core.formatter.insert_space_after_comma_in_enum_declarations"
                        value="insert" />
                <setting
                        id="org.eclipse.jdt.core.formatter.insert_space_before_comma_in_allocation_expression"
                        value="do not insert" />
                <setting
                        id="org.eclipse.jdt.core.formatter.parentheses_positions_in_for_statment"
                        value="common_lines" />
                <setting
                        id="org.eclipse.jdt.core.formatter.comment.new_lines_at_block_boundaries"
                        value="true" />
                <setting
                        id="org.eclipse.jdt.core.formatter.insert_space_after_comma_in_constructor_declaration_parameters"
                        value="insert" />
                <setting
                        id="org.eclipse.jdt.core.formatter.comment.insert_new_line_for_parameter"
                        value="do not insert" />
                <setting
                        id="org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_package"
                        value="insert" />
                <setting
                        id="org.eclipse.jdt.core.formatter.insert_space_between_empty_parens_in_enum_constant"
                        value="do not insert" />
                <setting
                        id="org.eclipse.jdt.core.formatter.insert_space_before_closing_paren_in_while"
                        value="do not insert" />
                <setting
                        id="org.eclipse.jdt.core.formatter.insert_space_between_empty_parens_in_annotation_type_member_declaration"
                        value="do not insert" />
                <setting
                        id="org.eclipse.jdt.core.formatter.comment.format_javadoc_comments"
                        value="false" />
                <setting id="org.eclipse.jdt.core.formatter.indentation.size"
```

gradle/eclipse-formatter-settings.xml +2 −2 View file @

```
... ...  @@ -127,7 +127,7 @@
127 127                      value="do not insert" />
128 128              <setting
129 129                      id="org.eclipse.jdt.core.formatter.insert_new_line_after_annotation_on_enum_constant"
130     -                    value="insert" />
    130 +                    value="do not insert" />
131 131              <setting
132 132                      id="org.eclipse.jdt.core.formatter.insert_space_after_multiplicative_operator"
133 133                      value="insert" />
... ...  @@ -259,7 +259,7 @@
259 259                      value="insert" />
260 260              <setting
261 261                      id="org.eclipse.jdt.core.formatter.alignment_for_annotations_on_enum_constant"
262     -                    value="0" />
    262 +                    value="48" />
263 263              <setting
264 264                      id="org.eclipse.jdt.core.formatter.insert_space_before_and_in_type_parameter"
265 265                      value="insert" />
... ...
```

DECISION DRIVERS	SPOTLESS
it is possible to automate code formatting	👍
convenient development experience	👍
migration should be smooth and manageable	👍
should support different file formats	⭐
code style should be similar to the informal one we already use, including the ability to fine-tune style to our needs	👍

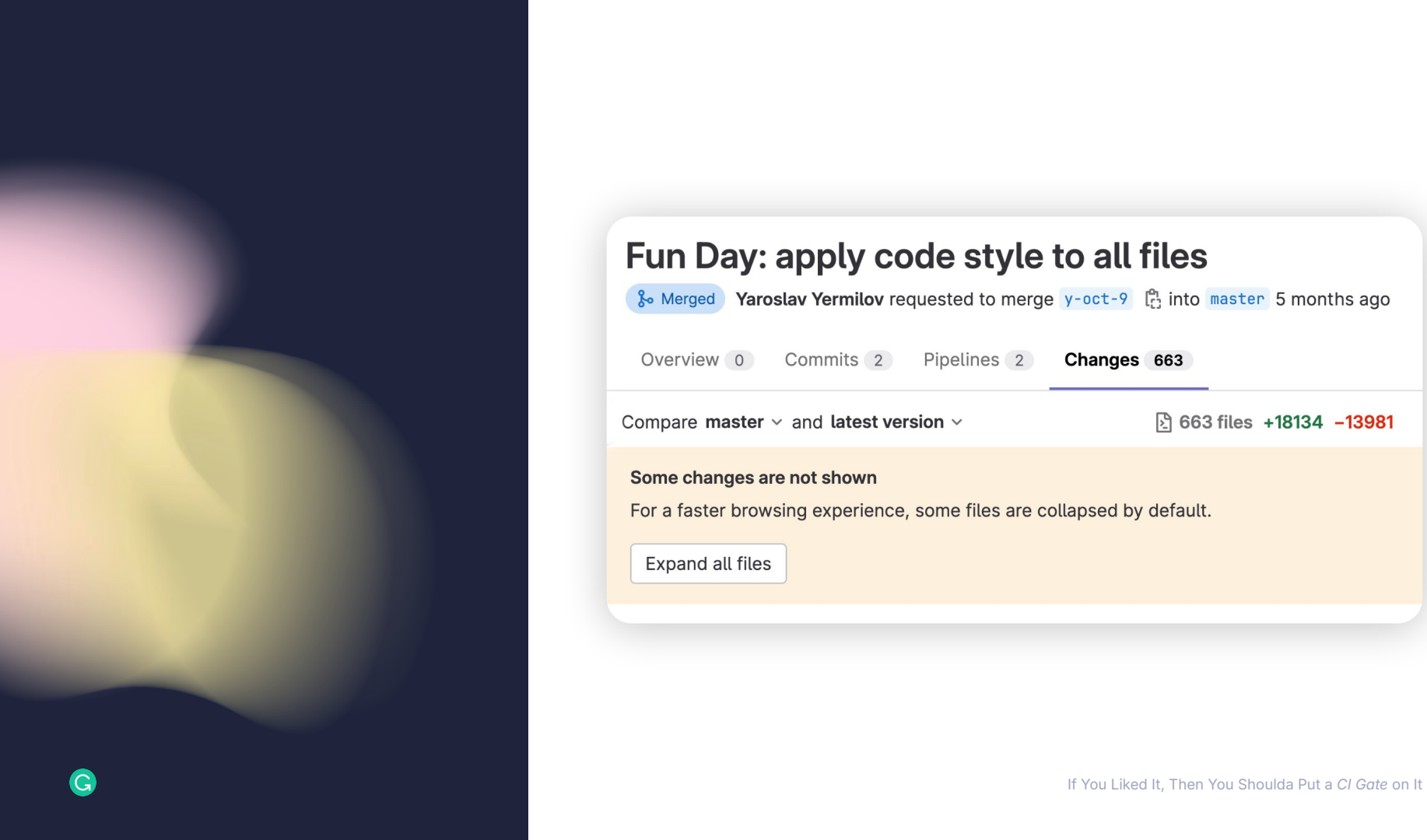
```
format 'terraform', {
    target 'terraform/**/*.tf', 'terraform/**/*.tfvars'

    toggleOffOn()

    String terraformPath = System.getenv('TERRAFORM_PATH') ?: '/usr/local/bin/terraform'
    if (new File(terraformPath).exists()) {
        nativeCmd('terraform', terraformPath, ['fmt', '-']) // name, path to binary, additional arguments
    }
}
```

```
format 'terraform', {
    target 'terraform/**/*.tf', 'terraform/**/*.tfvars'

    toggleOffOn()

    String terraformPath = System.getenv('TERRAFORM_PATH') ?: '/usr/local/bin/terraform'
    if (new File(terraformPath).exists()) {
        nativeCmd('terraform', terraformPath, ['fmt', '-']) // name, path to binary, additional arguments
    }
}

format 'xml', {
    target '**/*.xml'
    targetExclude 'tmp/**/*.xml', '.idea/**/*.xml', '**/test-results/**/*.xml'

    toggleOffOn()

    eclipseWtp('xml')
    endWithNewline()
}
```

```
format 'terraform', {
    target 'terraform/**/*.tf', 'terraform/**/*.tfvars'

    toggleOffOn()

    String terraformPath = System.getenv('TERRAFORM_PATH') ?: '/usr/local/bin/terraform'
    if (new File(terraformPath).exists()) {
        nativeCmd('terraform', terraformPath, ['fmt', '-']) // name, path to binary, additional arguments
    }
}

format 'xml', {
    target '**/*.xml'
    targetExclude 'tmp/**/*.xml', '.idea/**/*.xml', '**/test-results/**/*.xml'

    toggleOffOn()

    eclipseWtp('xml')
    endWithNewline()
}

json {
    target '**/*.json'
    targetExclude 'tmp/**/*.json', '**/.terraform/**/*.json', '**/bin/**/*.json', '**/denali*.json', '**/taskdef*.json', '**/cards/**/*.js

    toggleOffOn()

    gson()
            .indentWithSpaces(2)
            .version("${libs.versions.gson.get()}")
            .escapeHtml()
}
```

```
format 'terraform', {
    target 'terraform/**/*.tf', 'terraform/**/*.tfvars'

    toggleOffOn()

    String terraformPath = System.getenv('TERRAFORM_PATH') ?: '/usr/local/bin/terraform'
    if (new File(terraformPath).exists()) {
        nativeCmd('terraform', terraformPath, ['fmt', '-']) // name, path to binary, additional arguments
    }
}

format 'xml', {
    target '**/*.xml'
    targetExclude 'tmp/**/*.xml', '.idea/**/*.xml', '**/test-results/**/*.xml'

    toggleOffOn()

    eclipseWtp('xml')
    endWithNewline()
}

json {
    target '**/*.json'
    targetExclude 'tmp/**/*.json', '**/.terraform/**/*.json', '**/bin/**/*.json', '**/denali*.json', '**/taskdef*.json', '**/cards/**/*.js

    toggleOffOn()

    gson()
            .indentWithSpaces(2)
            .version("${libs.versions.gson.get()}")
            .escapeHtml()
}

format 'generic', {
    target '**/*.md', '.gitignore', '.dockerignore', '**/*.csv', '**/*.properties', '**/*.toml', '**/*.json5', '**/Dockerfile', '**/*.sh',
    targetExclude 'tmp/**/*', '**/.gradle/**/*'

    toggleOffOn()

    endWithNewline()
}
```

DECISION DRIVERS	SPOTLESS
it is possible to automate code formatting	👍
convenient development experience	👍
migration should be smooth and manageable	👍
should support different file formats	⭐
code style should be similar to the informal one we already use, including the ability to fine-tune style to our needs	👍


Fun Day: apply code style to all files
Merged
Yaroslav Yermilov requested to merge y-oct-9 into master 5 months ago
Overview 0
Commits 2
Pipelines 2
Changes 663
Compare master and latest version
663 files +18134 −13981
Some changes are not shown
For a faster browsing experience, some files are collapsed by default.
Expand all files

DECISION DRIVERS	SPOTLESS
it is possible to automate code formatting	👍
convenient development experience	👍
migration should be smooth and manageable	👍
should support different file formats	⭐
code style should be similar to the informal one we already use, including the ability to fine-tune style to our needs	👍

The *pre-commit* script consists of 3 parts.

1. **Assign all staged files to a variable:** This variable will contain the files, which the developer is about to commit. This is needed as all files with changes might not be staged.
2. **Apply the formatting:** This is the part where files might be altered during the formatting. *You could also run other plugin in addition to the formatting.*
3. **Re-add staged files:** All files that were identified in *Part 1* are staged with *git add*. A test is needed to check if the file exists, as a staged file might have been deleted.

```
#!/bin/sh

# Part 1
stagedFiles=$(git diff --staged --name-only)

# Part 2
echo "Running spotlessApply. Formatting code..."
./gradlew spotlessApply

# Part 3
for file in $stagedFiles; do
  if test -f "$file"; then
    git add $file
  fi
done
```

DECISION DRIVERS	SPOTLESS	SPOTLESS BUT WITH PRE-COMMIT HOOK
it is possible to automate code formatting	👍	😬
convenient development experience	👍	
migration should be smooth and manageable	👍	
should support different file formats	⭐	
code style should be similar to the informal one we already use, including the ability to fine-tune style to our needs	👍	

One way to automate this process is to rely on a Gradle task that you are often using.

1. Have a checked-in version of the *pre-commit* script
   e.g. *./scripts/pre-commit*
2. Now have Gradle copy this file into the *.git/hooks* folder one every execution.

```
tasks.register('updateGitHooks', Copy) {
    from './scripts/pre-commit'
    into './.git/hooks'
}
compileJava.dependsOn updateGitHooks
```

DECISION DRIVERS	SPOTLESS	SPOTLESS BUT WITH PRE-COMMIT HOOK
it is possible to automate code formatting	👍	😬
convenient development experience	👍	😬
migration should be smooth and manageable	👍	
should support different file formats	⭐	
code style should be similar to the informal one we already use, including the ability to fine-tune style to our needs	👍	

pre-commit.com

.pre-commit-config.yaml 268 bytes

```yaml
repos:
  - repo: local
    hooks:
      - id: spotless-apply
        name: Apply Spotless Formatting
        description: This hook applies Spotless formatting.
        entry: scripts/pre-commit/spotless-apply.sh
        language: system
        pass_filenames: false

```

```
#!/usr/bin/env bash

# heavily inspired by https://github.com/gruntwork-io/pre-commit/blob/master/hooks/terraform-fmt.sh

set -e

# OSX GUI apps do not pick up environment variables the same way as Terminal apps and there are no easy solutions,
# especially as Apple changes the GUI app behavior every release (see https://stackoverflow.com/q/135688/483528).
# As a workaround to allow OSX GUI to work, add this (hopefully harmless) setting here
original_path=$PATH
export PATH=$PATH:/usr/local/bin

# If JAVA_HOME is not set (because of the same problem as above) - hope that sdkman is installed
if [ -z "$JAVA_HOME" ] ; then
    export JAVA_HOME=~/.sdkman/candidates/java/current
fi

hook_error=0

./gradlew spotlessApply || hook_error=$?

# reset path to the original value
export PATH=$original_path

exit ${hook_error}
```

DECISION DRIVERS	SPOTLESS
we should be able to check if the formatting is correct from CI	👍
convenient development experience	👍
migration should be smooth and manageable	👍
should support different file formats	⭐
code style should be similar to the informal one we already use, including the ability to fine-tune style to our needs	👍

TL;DR

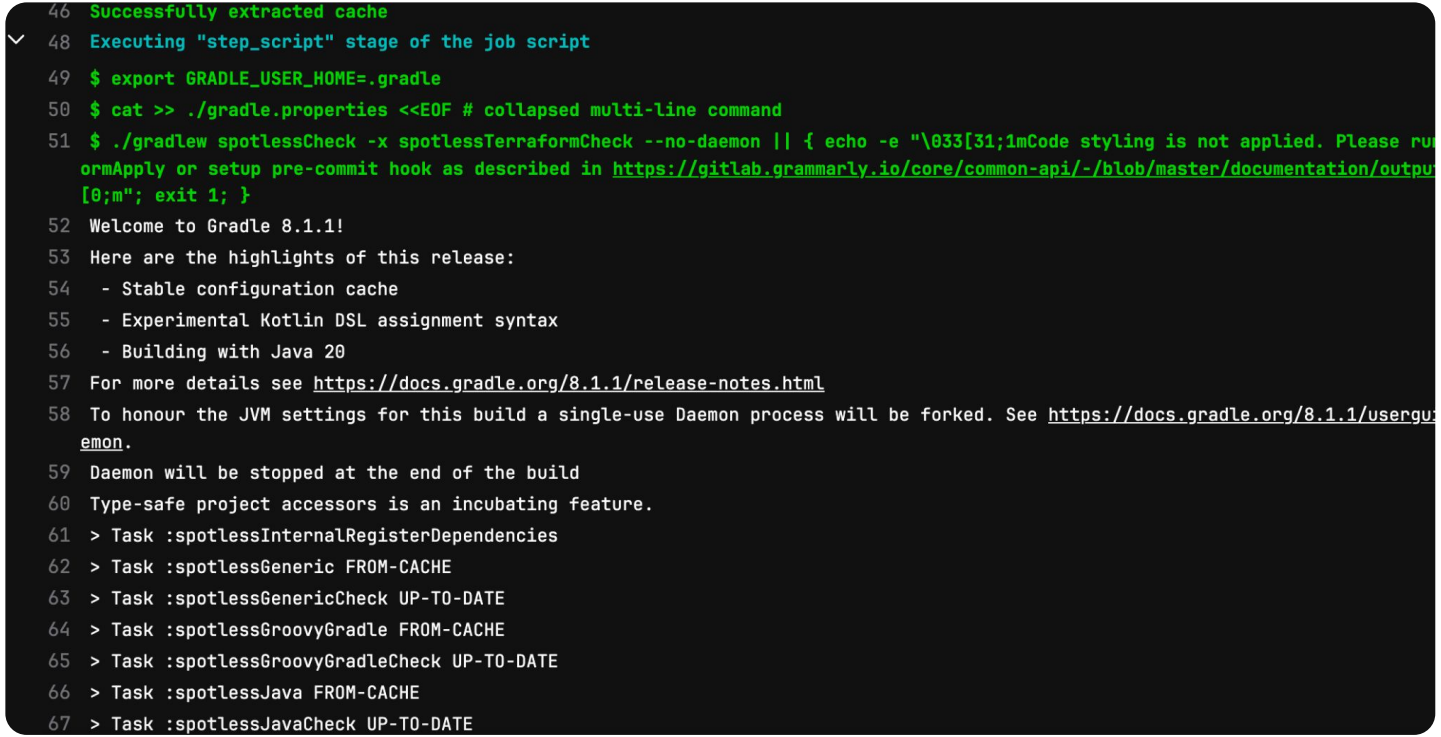
Choice is counter-productive

Don’t rely on engineer’s action

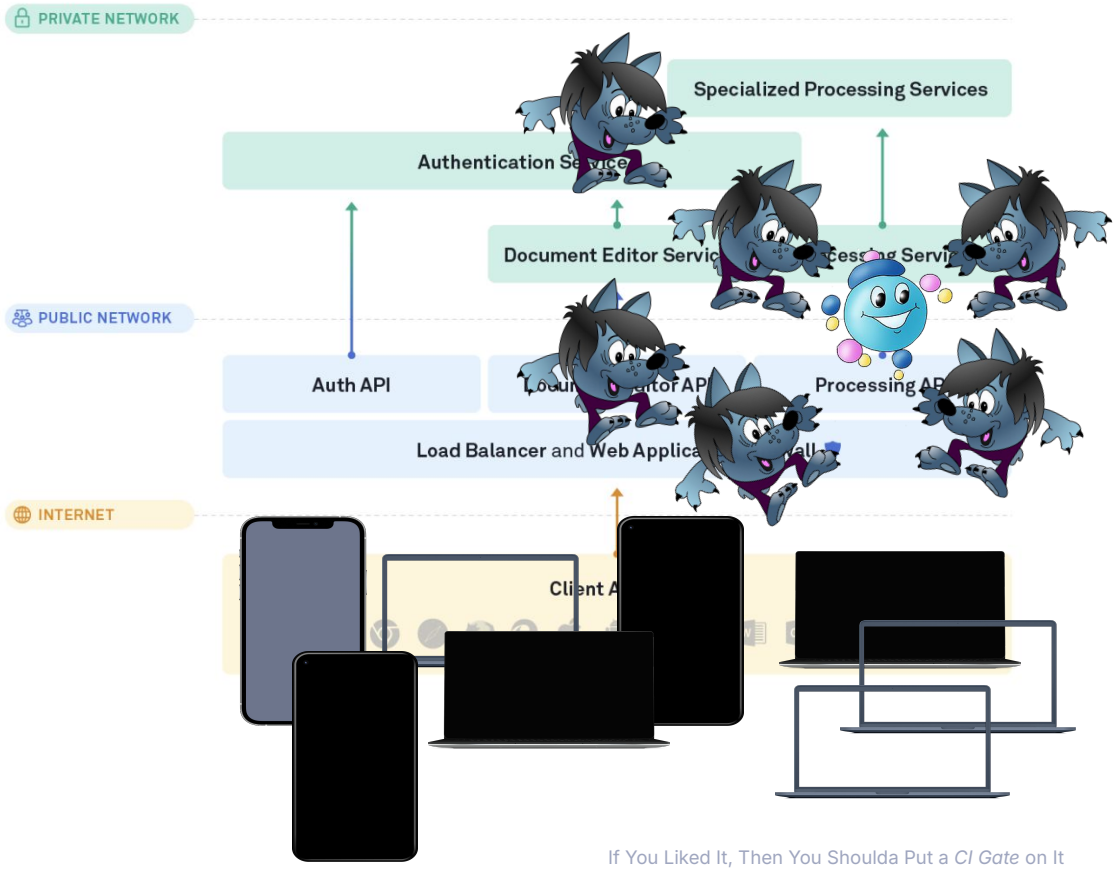
Think a lot about development experience

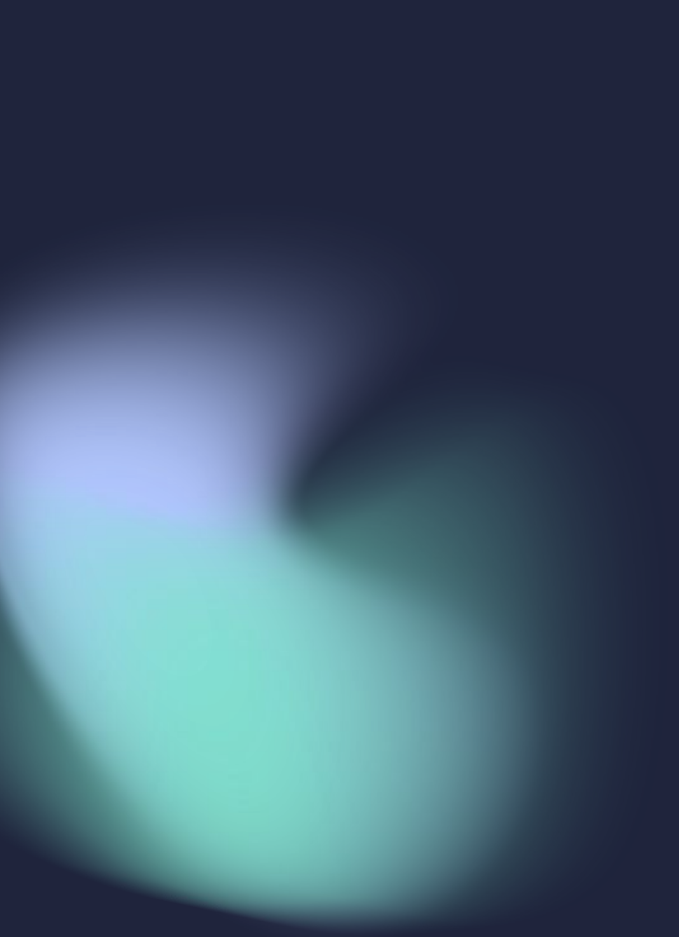
Be explicit about decisions

If You Liked It, Then You Shoulda Put a *CI Gate* on It

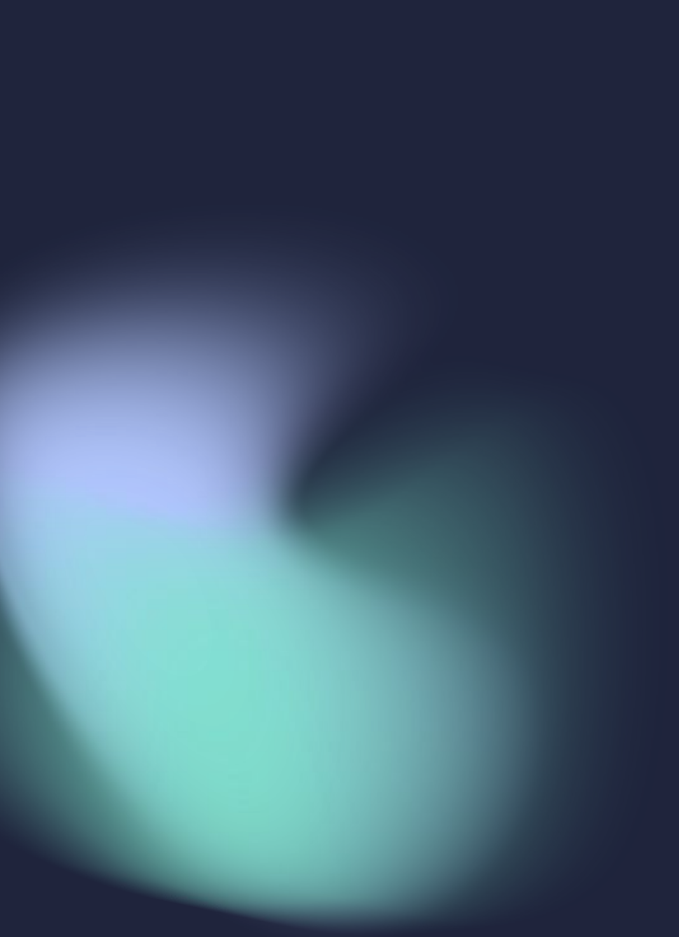

46 Successfully extracted cache
48 Executing "step_script" stage of the job script
49 $ export GRADLE_USER_HOME=.gradle
50 $ cat >> ./gradle.properties <<EOF # collapsed multi-line command
51 $ ./gradlew spotlessCheck -x spotlessTerraformCheck --no-daemon || { echo -e "\033[31;1mCode styling is not applied. Please ru
ormApply or setup pre-commit hook as described in https://gitlab.grammarly.io/core/common-api/-/blob/master/documentation/outpu
[0;m"; exit 1; }
52 Welcome to Gradle 8.1.1!
53 Here are the highlights of this release:
54 - Stable configuration cache
55 - Experimental Kotlin DSL assignment syntax
56 - Building with Java 20
57 For more details see https://docs.gradle.org/8.1.1/release-notes.html
58 To honour the JVM settings for this build a single-use Daemon process will be forked. See https://docs.gradle.org/8.1.1/usergu
emon.
59 Daemon will be stopped at the end of the build
60 Type-safe project accessors is an incubating feature.
61 > Task :spotlessInternalRegisterDependencies
62 > Task :spotlessGeneric FROM-CACHE
63 > Task :spotlessGenericCheck UP-TO-DATE
64 > Task :spotlessGroovyGradle FROM-CACHE
65 > Task :spotlessGroovyGradleCheck UP-TO-DATE
66 > Task :spotlessJava FROM-CACHE
67 > Task :spotlessJavaCheck UP-TO-DATE

Contributors from other teams





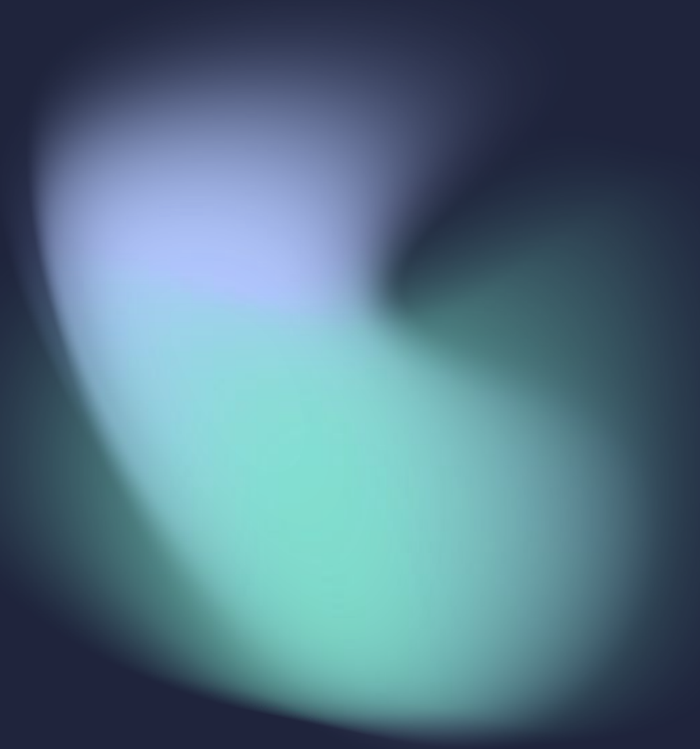
```
2023-05-14 14:13:57,584 [QueuedThreadPool-jetty-68] INFO
g.c.s.ClientInfo: [THROTTLED]x92 Client type received:
ClientInfo{clientType=EXTENSION_CHROME, clientSubtype=GENERAL,
clientVersion=0.0.0.0, clientSupports=[]}
2023-05-14 14:13:58,133 [ScheduledCheck-18] INFO
g.c.h.Sprawl: document check skipped alphabetic_ratio = 0.0
len = 248 fullText.len = 248
2023-05-14 14:13:58,135 [QueuedThreadPool-jetty-44] WARN
g.c.t.h.s.HttpRequestVerifier: CSRF checks failed
Reconnect failed: session 8281dd99-c052-4551-82fd-ba407e3230fc
reason no such session
2023-05-14 14:13:58,296 [QueuedThreadPool-jetty-60] INFO
g.c.b.a.AuthBackend: Authentication denied: NotFoundException:
HTTP 404 Not Found
```



```
2023-05-14 14:13:57,584 [QueuedThreadPool-jetty-68] INFO
g.c.s.ClientInfo: [THROTTLED]x92 Client type received:
ClientInfo{clientType=EXTENSION_CHROME, clientSubtype=GENERAL,
clientVersion=0.0.0.0, clientSupports=[]}
2023-05-14 14:13:58,133 [ScheduledCheck-18] INFO  g.c.h.Sprawl:
document check skipped alphabetic_ratio = 0.0 len = 248
fullText.len = 248
2023-05-14 14:13:58,135 [QueuedThreadPool-jetty-44] WARN
g.c.t.h.s.HttpRequestVerifier: CSRF checks failed
Reconnect failed: session 8281dd99-c052-4551-82fd-ba407e3230fc
reason no such session
2023-05-14 14:13:58,296 [QueuedThreadPool-jetty-60] INFO
g.c.b.a.AuthBackend: Authentication denied: NotFoundException:
HTTP 404 Not Found
```

```
115         );
116     }
117
-   public List<SprawlRequestBody> createSprawlSentenceRequest(SprawlRequest request, boolean useNewTokenizer) {
118 +   public List<SprawlRequestBody> createSprawlSentenceRequest(SprawlRequest request) {
119 +       System.out.println("request = " + request);
120         ParseMessage parseMessage = new ParseMessage(null);
121
122         parseMessage.setTexts(singletonList(request.text));
    @@ -123,10 +124,10 @@ public class CheckService {
124         parseMessage.setSplitIntoParagraphs(false);
125         parseMessage.setSplitIntoSentences(true);
126         parseMessage.setUseSpaceTokenizer(false);
-       parseMessage.setUseNewTokenizer(useNewTokenizer);
127         parseMessage.setRequestedFields(List.copyOf(ParseMessageHandler.availableRequestedFields));
128
129         Message result = parseMessageHandler.processParseMessage(parseMessage, ClientType.HTTP_PRIVATE_API);
130 +       System.out.println("result = " + result);
131
132         // Check for errors
133         if (result.getAction() == MessageType.ERROR) {
    @@ -155,7 +156,7 @@ public class CheckService {
156         }
157     }
```

TL;DR



Choice is counter-productive

Don’t rely on engineer’s action

Think a lot about development experience

Be explicit about decisions

If You Liked It, Then You Shoulda Put a *CI Gate* on It

github.com/policeman-tools/forbidden-apis

README.md

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).

Documentation

Please refer to the Github Wiki & Documentation.

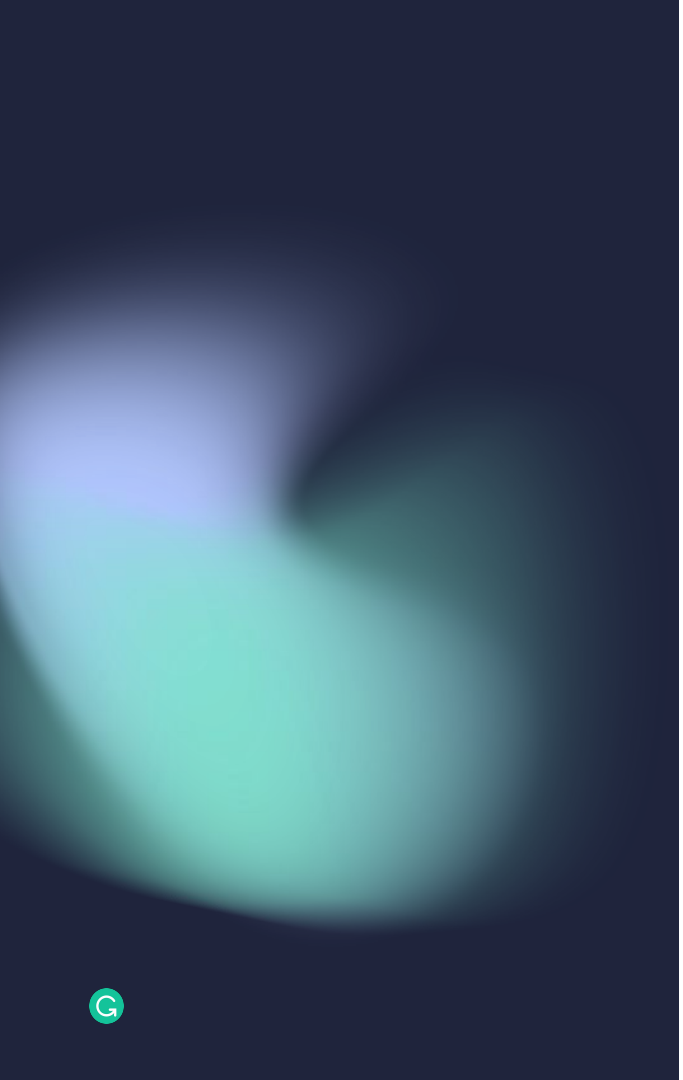
The checker is available as Apache Ant Task, Apache Maven Mojo, and Gradle plugin. In addition there is a command line tool (CLI):

- 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.

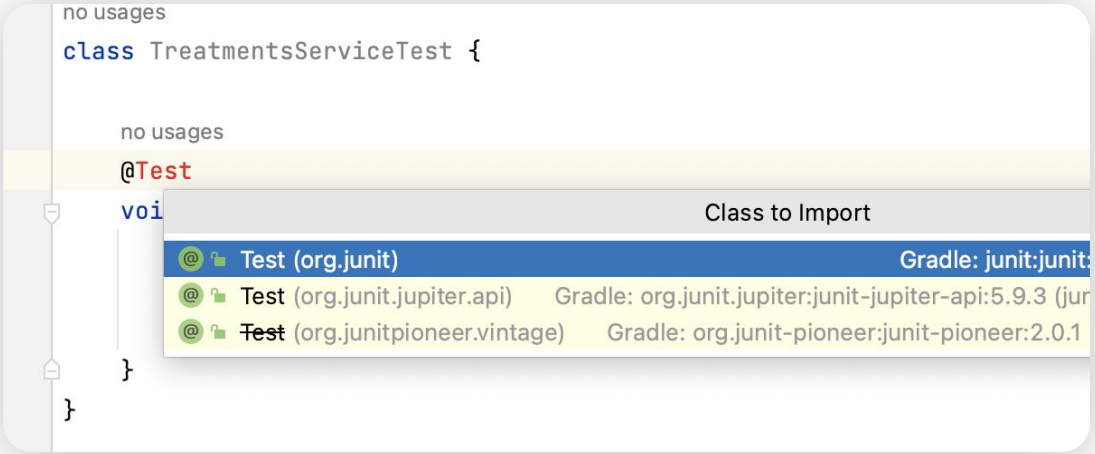
```
109  110    }
110  111
     112  + forbiddenApisMain {
     113  +     bundledSignatures = ['jdk-system-out']
     114  +     ignoreFailures = false
     115  + }
     116  +
111  117    test {
```

```
> Task :capi-server:forbiddenApisMain FAILED
Forbidden field access: java.lang.System#out [prints to System.out; should only be used for debugging, not in production code]
  in grammarly.capi.tools.DynamoScanAndReplace (DynamoScanAndReplace.java:40)
Forbidden method invocation: java.lang.Throwable#printStackTrace() [Eclipse auto-generated stubs; exceptions should be correctly
  in grammarly.capi.tools.DynamoScanAndReplace (DynamoScanAndReplace.java:56)
Forbidden method invocation: java.lang.Throwable#printStackTrace() [Eclipse auto-generated stubs; exceptions should be correctly
  in grammarly.capi.tools.DynamoScanAndReplace (DynamoScanAndReplace.java:67)
Forbidden method invocation: java.lang.Throwable#printStackTrace() [Eclipse auto-generated stubs; exceptions should be correctly
  in grammarly.capi.session.MessageHandler (MessageHandler.java:1778)
Forbidden field access: java.lang.System#out [prints to System.out; should only be used for debugging, not in production code]
  in grammarly.capi.Main (Main.java:130)
Forbidden field access: java.lang.System#out [prints to System.out; should only be used for debugging, not in production code]
  in grammarly.capi.metrics.MetricConsumer (MetricConsumer.java:32)
Forbidden field access: java.lang.System#out [prints to System.out; should only be used for debugging, not in production code]
  in grammarly.capi.transport.http.base.CheckService (CheckService.java:117)
Forbidden field access: java.lang.System#out [prints to System.out; should only be used for debugging, not in production code]
  in grammarly.capi.transport.http.base.CheckService (CheckService.java:128)
Scanned 1210 class file(s) for forbidden API invocations (in 0.72s), 8 error(s).
FAILURE: Build failed with an exception.
```

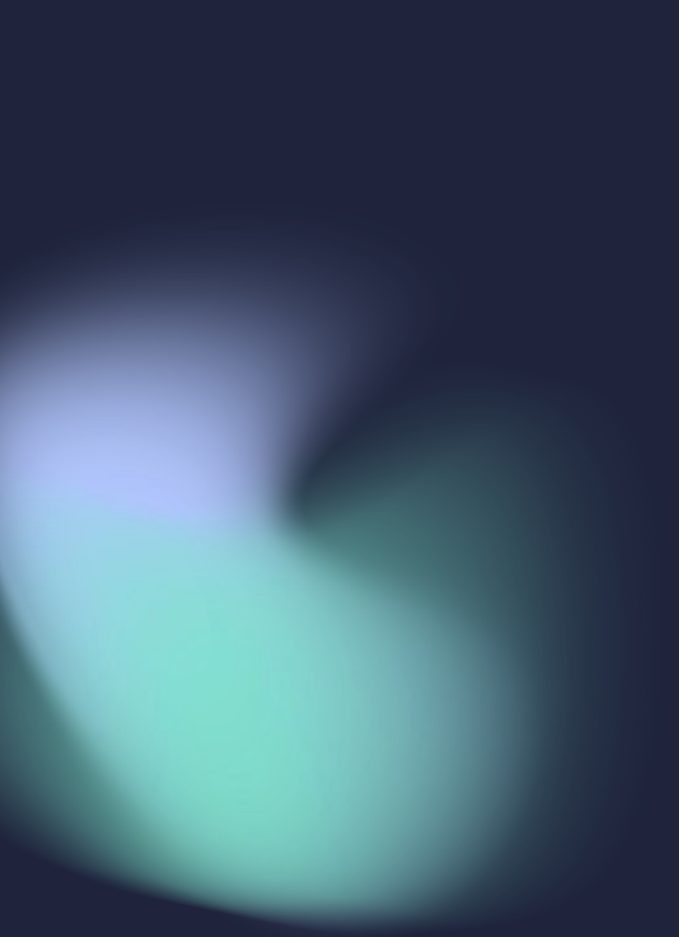


```
+ import java.time.LocalDate;
+ import java.time.Month;
+ import java.util.Set;
+
+ import static grammarly.capi.message.ClientType.EXTENSION_CHROME;
+ import static grammarly.capi.test.SessionBuilder.session;
+ import static grammarly.capi.test.verifier.AlertsVerifiers.*;
+ import static grammarly.capi.test.verifier.AlertsVerifiers.withExtraProperty;
+ import static grammarly.capi.test.verifier.ListVerifier.*;
+ import static grammarly.capi.test.verifier.PropertyVerifier.valueOfPropertyEqualsTo;
+ import static grammarly.capi.test.verifier.RecordVerifier.ofType;
+
+ @Execution(ExecutionMode.CONCURRENT)
+ public class PersonalizationAutoapplyTest extends CapiTest {
+
+     @Test
+     public void personalization_autoapply_internal() {
+         // when
+         user(
+                 withEmail(
+                         "yaroslav.yermilov@grammarly.com",
```

```
5  + import org.junit.Test;
6  + import org.junit.jupiter.api.parallel.Execution;
7  + import org.junit.jupiter.api.parallel.ExecutionMode;
8  +
9  + import java.time.LocalDate;
10 + import java.time.Month;
11 + import java.util.Set;
12 +
13 + import static grammarly.capi.message.ClientType.EXTENSION_CHROME;
14 + import static grammarly.capi.test.SessionBuilder.session;
15 + import static grammarly.capi.test.verifier.AlertsVerifiers.*;
16 + import static grammarly.capi.test.verifier.AlertsVerifiers.withExtraProperty;
17 + import static grammarly.capi.test.verifier.ListVerifier.*;
18 + import static grammarly.capi.test.verifier.PropertyVerifier.valueOfPropertyEqualsTo;
19 + import static grammarly.capi.test.verifier.RecordVerifier.ofType;
20 +
21 + @Execution(ExecutionMode.CONCURRENT)
22 + public class PersonalizationAutoapplyTest extends CapiTest {
23 +
24 +     @Test
25 +     public void personalization_autoapply_internal() {
26 +         // when
27 +         user(
28 +                 withEmail(
29 +                         "yaroslav.yermilov@grammarly.com",
```


no usages
class TreatmentsServiceTest {
no usages
@Test
voi
Class to Import
Test (org.junit) Gradle: junit:junit
Test (org.junit.jupiter.api) Gradle: org.junit.jupiter:junit-jupiter-api:5.9.3 (jur
Test (org.junitpioneer.vintage) Gradle: org.junit-pioneer:junit-pioneer:2.0.1
}
}

```
157
158 forbiddenApis {
159     signaturesFiles = files('../gradle/forbiddenapis.txt')
160     ignoreFailures = false
161 }
162
163 forbiddenApisMain {
164     bundledSignatures = ['jdk-system-out']
165     ignoreFailures = false
166 }
167
168 forbiddenApisTest {
169     signaturesFiles = files('../gradle/forbiddenapis-test.txt')
170     ignoreFailures = false
171 }
172
173 forbiddenApisIntegrationTest {
174     signaturesFiles = files('../gradle/forbiddenapis-test.txt')
175     ignoreFailures = false
176 }
177
```

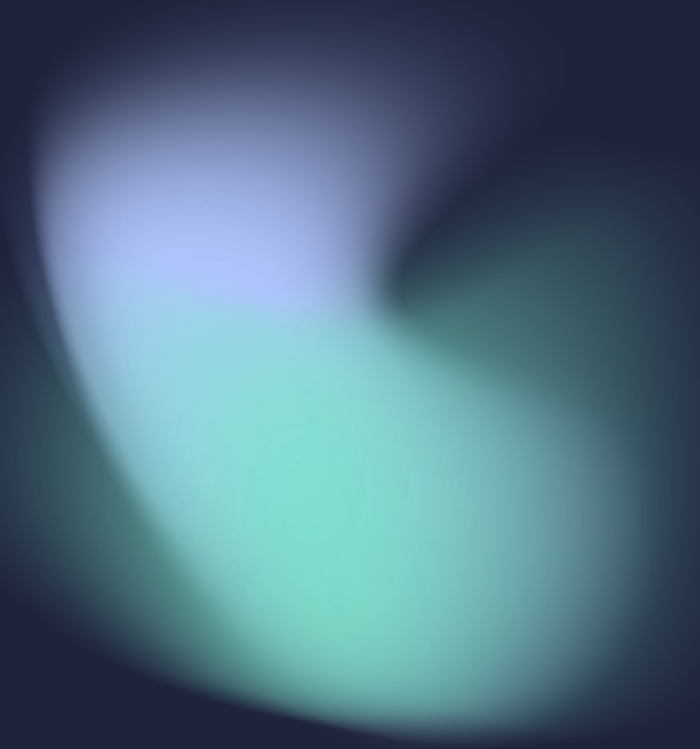


master common-api / gradle / forbiddenapis-test.txt

forbiddenapis-test.txt 56 bytes

```
1 org.junit.Test @ Use org.junit.jupiter.api.Test instead
2
```

TL;DR



Choice is counter-productive

Don’t rely on engineer’s action

Think a lot about development experience

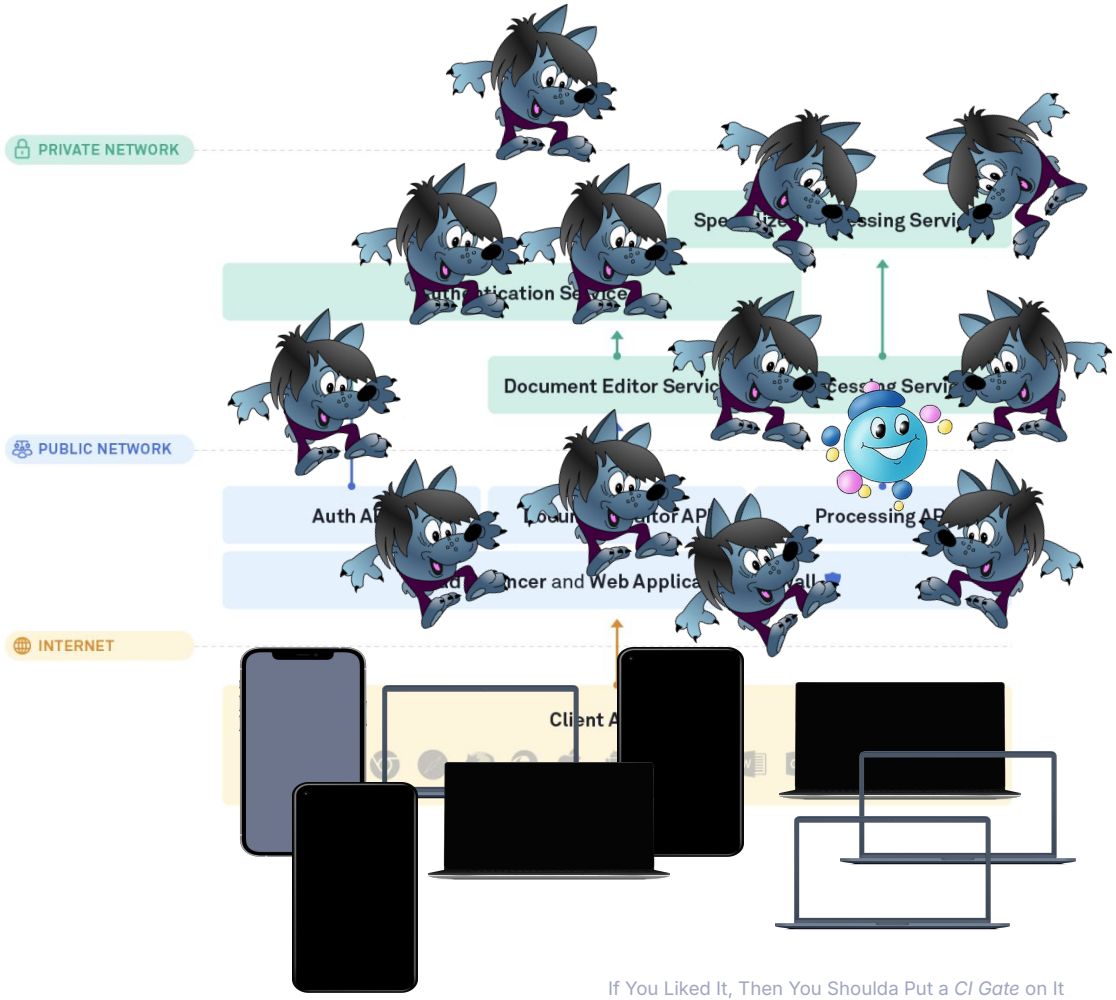
Be explicit about decisions

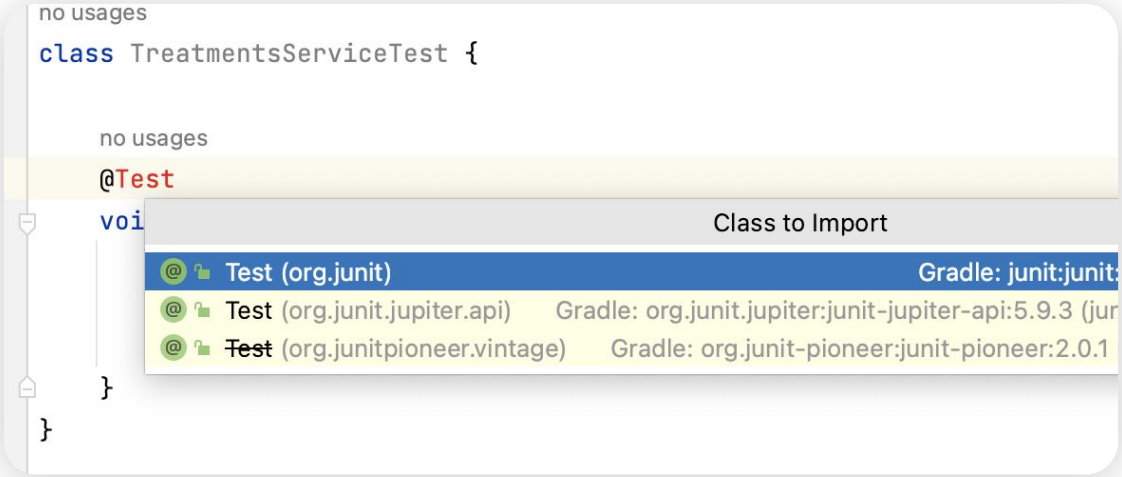
If You Liked It, Then You Shoulda Put a *CI Gate* on It

```
$ export GRADLE_USER_HOME=.gradle
$ cat >> ./gradle.properties <<EOF # collapsed multi-line command
$ ./gradlew forbiddenApis --no-daemon
Welcome to Gradle 8.1.1!
Here are the highlights of this release:
 - Stable configuration cache
 - Experimental Kotlin DSL assignment syntax
 - Building with Java 20
For more details see https://docs.gradle.org/8.1.1/release-notes.html
To honour the JVM settings for this build a single-use Daemon process will be forked.
emon.
Daemon will be stopped at the end of the build
Type-safe project accessors is an incubating feature.
> Task :capi-lib:compileJava FROM-CACHE
> Task :capi-lib:processResources NO-SOURCE
> Task :capi-lib:classes UP-TO-DATE
> Task :capi-lib:jar
> Task :configuration-api:compileJava FROM-CACHE
> Task :configuration-api:processResources NO-SOURCE
```

```
$ export GRADLE_USER_HOME=.gradle
$ cat >> ./gradle.properties <<EOF # collapsed multi-line command
$ ./gradlew forbiddenApis --no-daemon
Welcome to Gradle 8.1.1!
Here are the highlights of this release:
 - Stable configuration cache
 - Experimental Kotlin DSL assignment syntax
 - Building with Java 20
For more details see https://docs.gradle.org/8.1.1/release-notes.html
To honour the JVM settings for this build a single-use Daemon process will be forked.
emon.
Daemon will be stopped at the end of the build
Type-safe project accessors is an incubating feature.
> Task :capi-lib:compileJava FROM-CACHE
> Task :capi-lib:processResources NO-SOURCE
> Task :capi-lib:classes UP-TO-DATE
> Task :capi-lib:jar
> Task :configuration-api:compileJava FROM-CACHE
> Task :configuration-api:processResources NO-SOURCE
```

500 MR from 50 contributors




no usages
class TreatmentsServiceTest {
no usages
@Test
voi
Class to Import
Test (org.junit) Gradle: junit:junit
Test (org.junit.jupiter.api) Gradle: org.junit.jupiter:junit-jupiter-api:5.9.3 (jur
Test (org.junitpioneer.vintage) Gradle: org.junit-pioneer:junit-pioneer:2.0.1
}
}

TL;DR

Choice is counter-productive

Don’t rely on engineer’s action

Think a lot about development experience

Be explicit about decisions

If You Liked It, Then You Shoulda Put a *CI Gate* on It

README.asciidoc

release v1.20.0
snapshot v1.20.1-SNAPSHOT
Main passing

Detect unused and misused dependencies

The Dependency Analysis Gradle Plugin (DAGP, née Dependency Analysis Android Gradle Plugin) detects the following:

1. Unused dependencies.
2. Used transitive dependencies (which you may want to declare directly).
3. Dependencies declared on the wrong configuration (`api` vs `implementation` vs `compileOnly`, etc.).

github.com/autonomousapps/dependency-analysis-android-gradle-plugin

Add to your project and use

For detailed instructions, see the wiki.

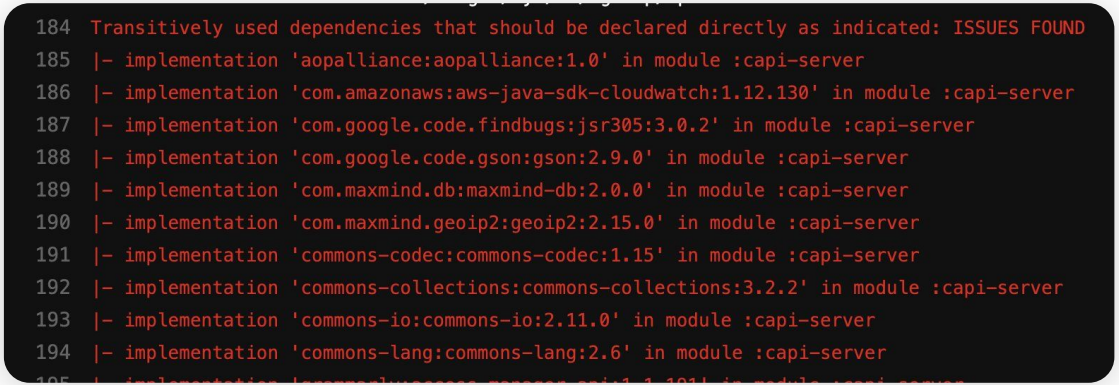
The simplest approach is to add the following:

root build.gradle

```
plugins {
  id("com.autonomousapps.dependency-analysis") version "<<latest_version>>"
}
```

For a quick start, just run the following:

```
./gradlew buildHealth
```


184 Transitively used dependencies that should be declared directly as indicated: ISSUES FOUND
185 |- implementation 'aopalliance:aopalliance:1.0' in module :capi-server
186 |- implementation 'com.amazonaws:aws-java-sdk-cloudwatch:1.12.130' in module :capi-server
187 |- implementation 'com.google.code.findbugs:jsr305:3.0.2' in module :capi-server
188 |- implementation 'com.google.code.gson:gson:2.9.0' in module :capi-server
189 |- implementation 'com.maxmind.db:maxmind-db:2.0.0' in module :capi-server
190 |- implementation 'com.maxmind.geoip2:geoip2:2.15.0' in module :capi-server
191 |- implementation 'commons-codec:commons-codec:1.15' in module :capi-server
192 |- implementation 'commons-collections:commons-collections:3.2.2' in module :capi-server
193 |- implementation 'commons-io:commons-io:2.11.0' in module :capi-server
194 |- implementation 'commons-lang:commons-lang:2.6' in module :capi-server

TL;DR

Choice is counter-productive

Don’t rely on engineer’s action

Think a lot about development experience

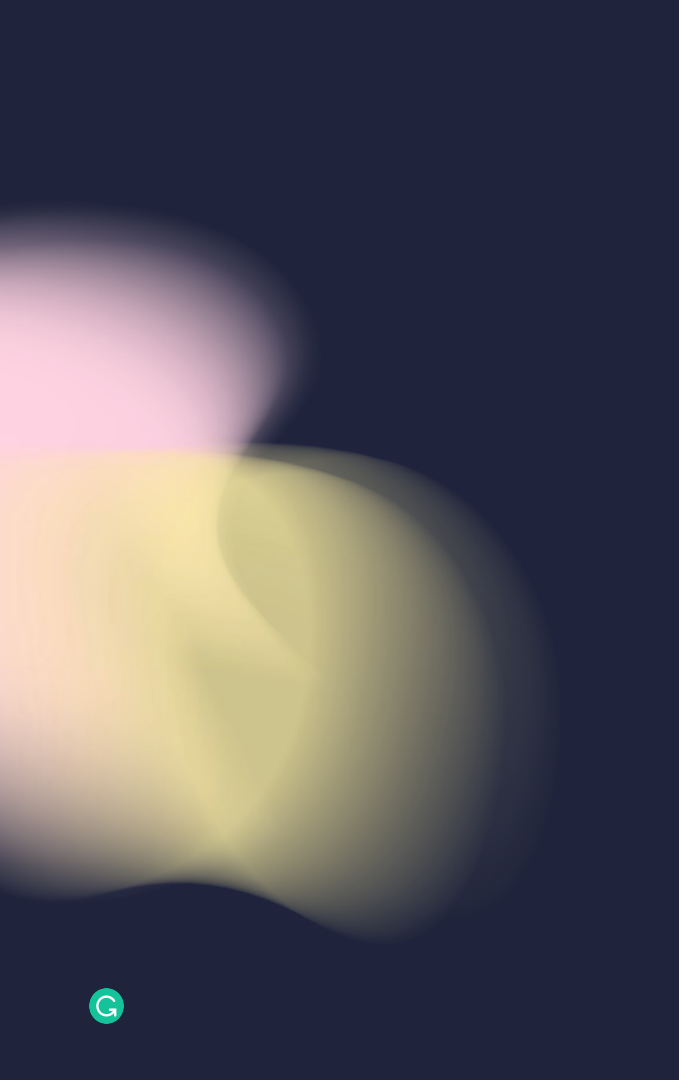
Be explicit about decisions

If You Liked It, Then You Shoulda Put a *CI Gate* on It

Forcing things explicit helps to avoid mistakes

```
exclude module: "slf4j-simple"
exclude module: "slf4j-timbre"
exclude module: "spring-boot-starter-logging"

resolutionStrategy {
    failOnVersionConflict()

    if (!gradle.startParameter.taskRequests.any({ it.args.contains('dependencyUpdates') })) {
        force "ch.qos.logback:logback-classic:${libs.versions.logback.get()}"
        force "ch.qos.logback:logback-core:${libs.versions.logback.get()}"
        force "com.amazonaws:aws-java-sdk-cloudwatch:${libs.versions.awsSdk1.get()}"
        force "com.amazonaws:aws-java-sdk-core:${libs.versions.awsSdk1.get()}"
        force "com.amazonaws:aws-java-sdk-dynamodb:${libs.versions.awsSdk1.get()}"
        force "com.amazonaws:aws-java-sdk-glue:${libs.versions.awsSdk1.get()}"
        force "com.amazonaws:aws-java-sdk-kms:${libs.versions.awsSdk1.get()}"
        force "com.amazonaws:aws-java-sdk-s3:${libs.versions.awsSdk1.get()}"
```

```
Unused dependencies which should be removed: ISSUES FOUND
|- implementation 'ch.qos.logback.contrib:logback-jackson:0.1.5' in module :capi-server
|- implementation 'ch.qos.logback.contrib:logback-json-classic:0.1.5' in module :capi-server
|- implementation 'com.amazonaws:amazon-kinesis-client:1.14.8' in module :capi-server
|- implementation 'com.amazonaws:jmespath-java:1.12.211' in module :capi-server
|- implementation 'com.auth0:jwks-rsa:0.21.1' in module :capi-server
|- implementation 'com.google.inject.extensions:guice-servlet:5.1.0' in module :capi-server
|- implementation 'io.confluent:kafka-avro-serializer:7.1.1' in module :capi-server
|- implementation 'jakarta.annotation:jakarta.annotation-api:2.1.0' in module :capi-server
|- implementation 'jakarta.xml.bind:jakarta.xml.bind-api:4.0.0' in module :capi-server
|- implementation 'javax.activation:activation:1.1.1' in module :capi-server
|- implementation 'javax.annotation:javax.annotation-api:1.3.2' in module :capi-server
|- implementation 'javax.xml.bind:jaxb-api:2.3.1' in module :capi-server
|- implementation 'net.jcip:jcip-annotations:1.0' in module :capi-server
```



Update core-docker.artifactory.grammarly.io/memcached Docker tag to v1.6.19 0 of 1 checklist item completed
!7865 · created 3 weeks ago by Sec Automation
renovate

Update dependency @easyops-cn/docusaurus-search-local to v0.35.0 0 of 1 checklist item completed
!8071 · created 6 days ago by Sec Automation
renovate

Update Grammarly internal API packages 0 of 1 checklist item completed
!8075 · created 6 days ago by Sec Automation
renovate

Update AWS SDK packages 0 of 1 checklist item completed
!8074 · created 6 days ago by Sec Automation
renovate

Update dependency com.google.protobuf to v0.9.3 0 of 1 checklist item completed
!8053 · created 6 days ago by Sec Automation
renovate

Update core-docker.artifactory.grammarly.io/hashicorp/terraform Docker tag to v1.4.6 0 of 1 checklist item completed
!8052 · created 1 week ago by Sec Automation
renovate

Update junit5 monorepo 0 of 1 checklist item completed
!8040 · created 1 week ago by Sec Automation
renovate

Update Grammarly internal API packages 0 of 1 checklist item completed
!8051 · created 1 week ago by Sec Automation
renovate

Update dependency com.google.api.grpc:proto-google-common-protos to v2.17.0 0 of 1 checklist item completed
!8062 · created 6 days ago by Sec Automation
renovate

TL;DR

Choice is counter-productive

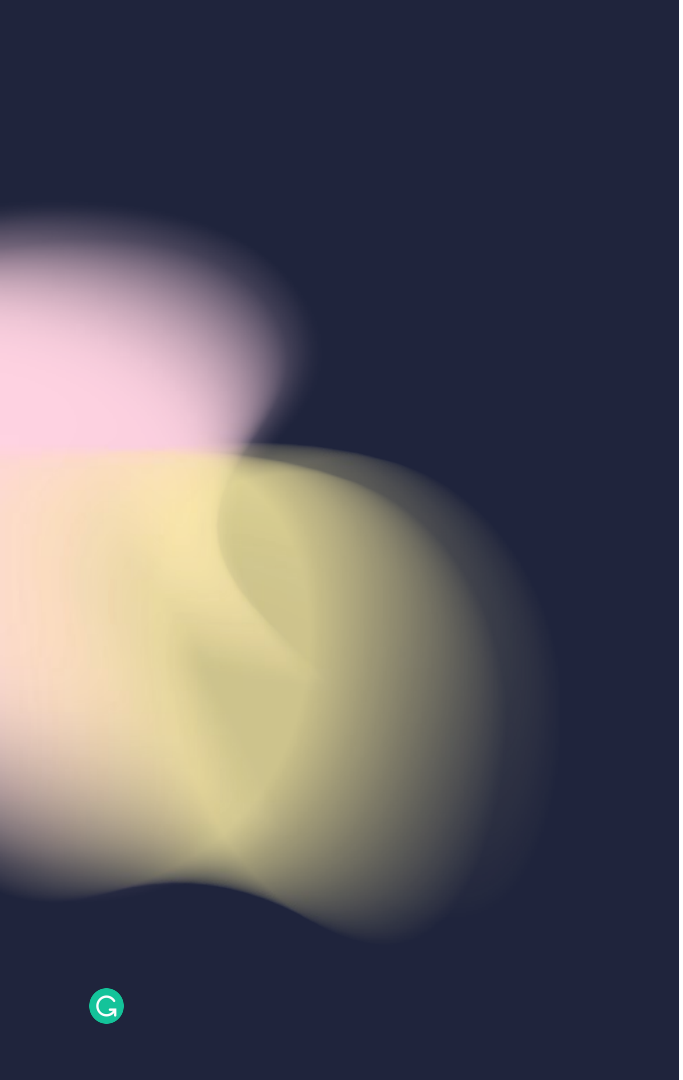
Don’t rely on engineer’s action

Think a lot about development experience

Be explicit about decisions

If You Liked It, Then You Shoulda *Automate* It

Forcing things explicit helps to avoid mistakes



Update core-docker.artifactory.grammarly.io/confluentinc/cp-kafka Docker tag to v7.4.0

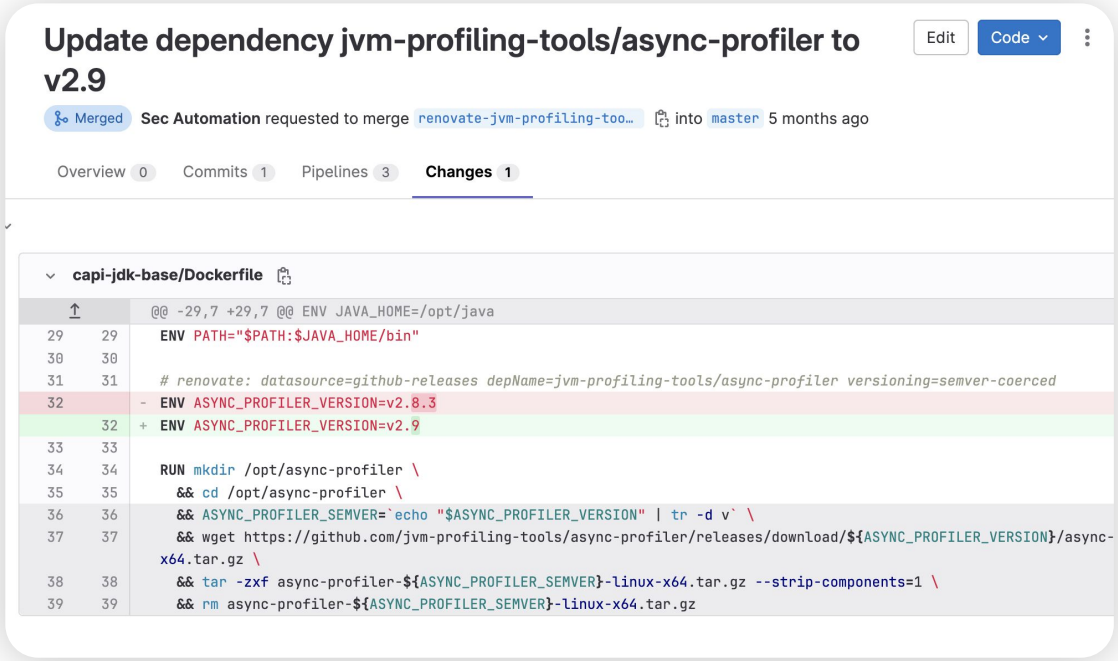
Edit

Open **Sec Automation** requested to merge `renovate-core-docker.artif…` into `master` 20 hours ago

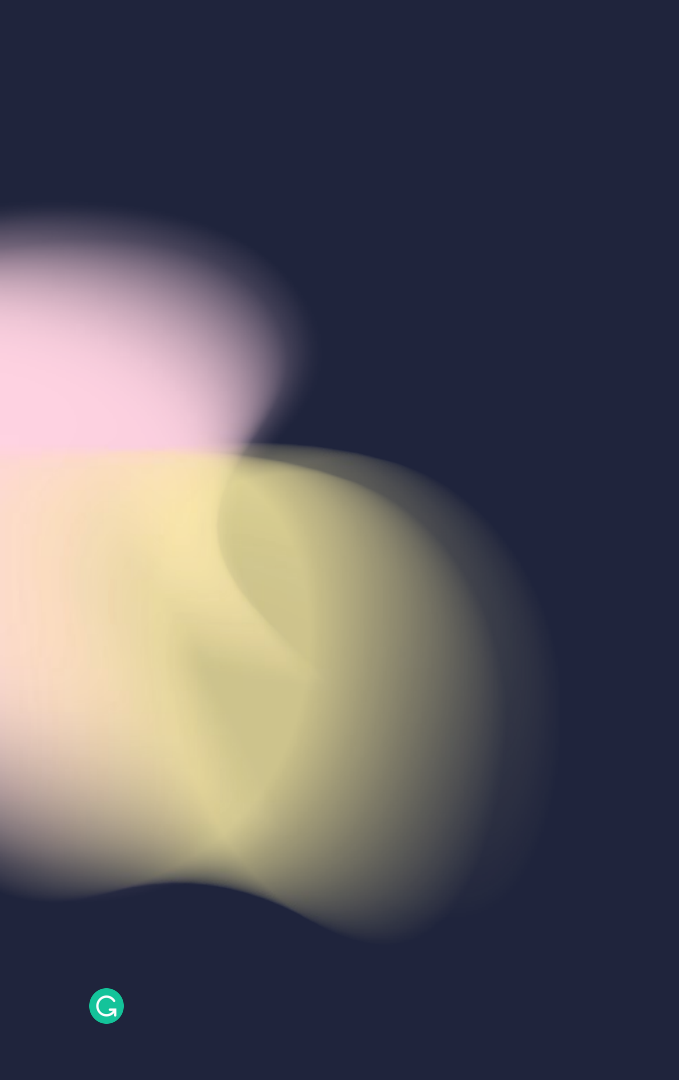
Overview 1 | Commits 1 | Pipelines 7 | **Changes 1**

capi-server/src/it/java/it/InstitutionUpdateKafkaListenerTest.java

```
@@ -55,7 +55,7 @@ public class InstitutionUpdateKafkaListenerTest {
55 55      public static final String email = "user@email.com";
56 56
57 57      public static final DockerImageName KAFKA_IMAGE_NAME = DockerImageName
58    -             .parse("core-docker.artifactory.grammarly.io/confluentinc/cp-kafka:7.3.3")
   58 +             .parse("core-docker.artifactory.grammarly.io/confluentinc/cp-kafka:7.4.0")
59 59              .asCompatibleSubstituteFor("confluentinc/cp-kafka");
60 60
61 61      protected static final KafkaContainer kafkaContainer = new KafkaContainer(KAFKA_IMAGE_NAME);
```


Update dependency jvm-profiling-tools/async-profiler to v2.9
Edit
Code
Merged
Sec Automation requested to merge renovate-jvm-profiling-too…
into master 5 months ago
Overview 0
Commits 1
Pipelines 3
Changes 1
capi-jdk-base/Dockerfile
@@ -29,7 +29,7 @@ ENV JAVA_HOME=/opt/java
ENV PATH="$PATH:$JAVA_HOME/bin"
renovate: datasource=github-releases depName=jvm-profiling-tools/async-profiler versioning=semver-coerced
- ENV ASYNC_PROFILER_VERSION=v2.8.3
+ ENV ASYNC_PROFILER_VERSION=v2.9
RUN mkdir /opt/async-profiler \
&& cd /opt/async-profiler \
&& ASYNC_PROFILER_SEMVER=`echo "$ASYNC_PROFILER_VERSION" | tr -d v` \
&& wget https://github.com/jvm-profiling-tools/async-profiler/releases/download/${ASYNC_PROFILER_VERSION}/async-profiler-${ASYNC_PROFILER_SEMVER}-linux-x64.tar.gz \
&& tar -zxf async-profiler-${ASYNC_PROFILER_SEMVER}-linux-x64.tar.gz --strip-components=1 \
&& rm async-profiler-${ASYNC_PROFILER_SEMVER}-linux-x64.tar.gz

```
"regexManagers": [
  {
    "fileMatch": [".*Dockerfile$"],
    "matchStrings": [
      "datasource=(?<datasource>.*?) depName=(?<depName>.*?)( versioning=(?<versioning>.*?))?\\sE
    ],
    "versioningTemplate": "{{#if versioning}}{{{versioning}}}{{else}}semver{{/if}}"
  },
  {
    "fileMatch": [".*java$"],
    "matchStrings": [
      ".*DockerImageName\\s*\\.parse\\(\"(?<depName>.*?):(?<currentValue>.*?)\"\\).*"
    ],
    "datasourceTemplate": "docker"
  }
],
```



```
[versions]
android-json = "0.0.20131108.vaadin1"
antlr = "4.9.3" # Caution! Previous update caused PIR: https://gitlab.grammarly.io/core/c
aopalliance = "1.0"
archaius = "0.7.7"
asm = "9.5"
aspectjweaver = "1.9.19"
assertj = "3.24.2"
autoValue = "1.10.1"
avro = "1.11.1"
awsElasticache = "1.2.0"
awsEncryptionSdk = "2.4.0"
awsGlue = "1.1.15"
awsKinesisProducer = "0.15.7"
awsMskIamAuth = "1.1.6"
awsRedshift = "1.2.55.1083"
awsSdk1 = "1.12.461"
awsSdk2 = "2.20.58"
benManesVersions = "0.46.0"
bouncycastleFips = "1.0.2.3"
bytebuddy = "1.14.4"
checkerQual = "3.34.0"
cheetahClientProto = "11.3.0"
clojure = "1.11.1"
commonmark = "0.17.0"
commonsCodec = "1.15"
commonsCollections = "3.2.2"
commonsCompress = "1.23.0"
commonsConfiguration = "1.10"
commonsIo = "2.11.0"
commonsLang2 = "2.6"
commonsLang3 = "3.12.0"
commonsMath = "3.6.1"
commonsText = "1.10.0"
commonsValidator = "1.7"
```

TL;DR

Choice is counter-productive

Don’t rely on engineer’s action

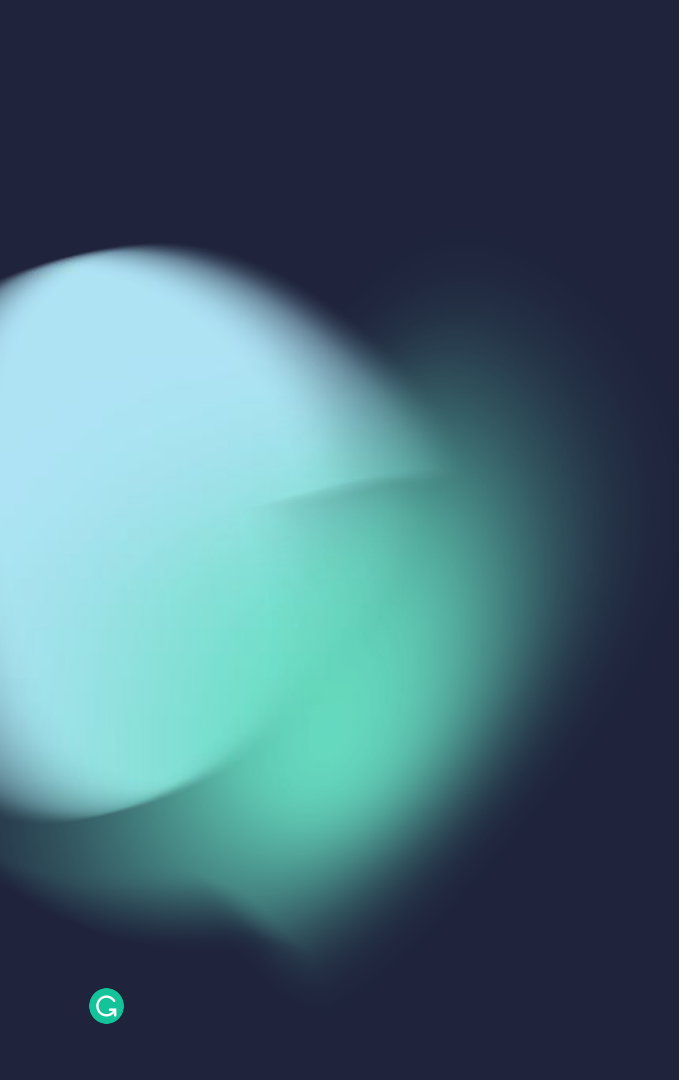
Think a lot about development experience

Be explicit about decisions

If You Liked It, Then You Shoulda Put a *CI Gate* on It

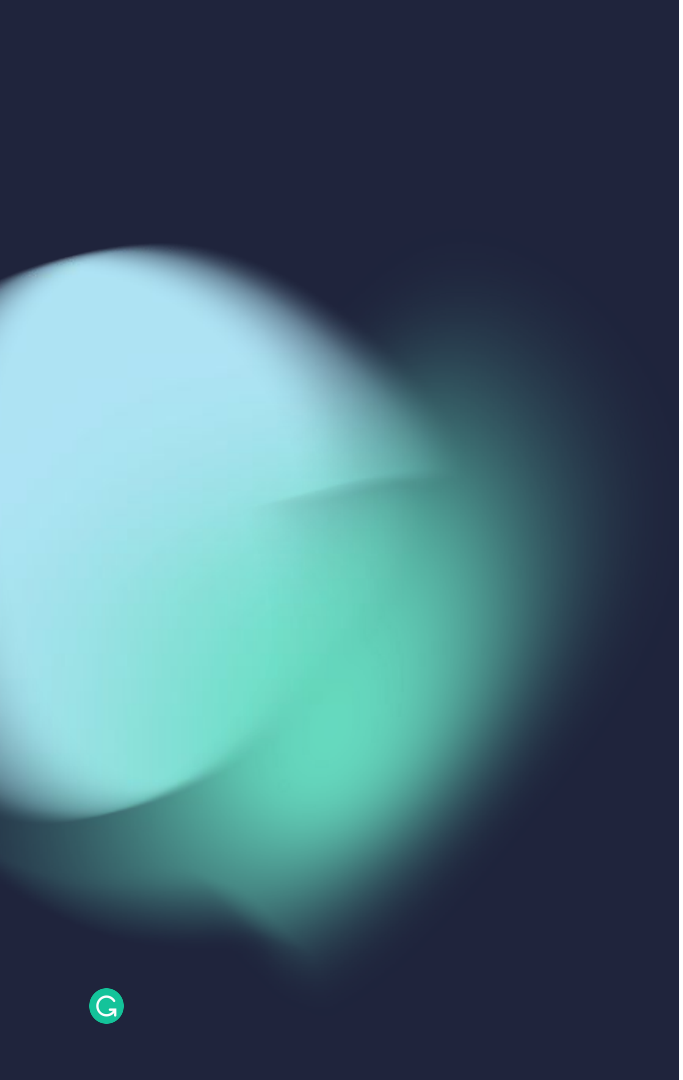
Forcing things explicit helps to avoid mistakes

```
[libraries]
android-json = { module = "com.vaadin.external.google:android-json", version = { ref = "android-json" } }
antlr4-runtime = { module = "org.antlr:antlr4-runtime", version = { ref = "antlr" } }
aopalliance = { module = "aopalliance:aopalliance", version = { ref = "aopalliance" } }
archaius = { module = "com.netflix.archaius:archaius-core", version = { ref = "archaius" } }
asm = { module = "org.ow2.asm:asm", version = { ref = "asm" } }
aspectjweaver = { module = "org.aspectj:aspectjweaver", version = { ref = "aspectjweaver" } }
assertj-core = { module = "org.assertj:assertj-core", version = { ref = "assertj" } }
autoValue-annotations = { module = "com.google.auto.value:auto-value-annotations", version = { ref = "autoValue" } }
avro = { module = "org.apache.avro:avro", version = { ref = "avro" } }
awsElasticache = { module = "com.amazonaws:elasticache-java-cluster-client", version = { ref = "awsElasticache" } }
awsGlue-registry = { module = "software.amazon.glue:schema-registry-common", version = { ref = "awsGlue" } }
awsGlue-registry-serde = { module = "software.amazon.glue:schema-registry-serde", version = { ref = "awsGlue" } }
awsKinesis-producer = { module = "com.amazonaws:amazon-kinesis-producer", version = { ref = "awsKinesisProducer" } }
awsMsk-iam-auth = { module = "software.amazon.msk:aws-msk-iam-auth", version = { ref = "awsMskIamAuth" } }
awsRedshift = { module = "com.amazon.redshift:redshift-jdbc42-no-awssdk", version = { ref = "awsRedshift" } }
awsSdk1-cloudwatch = { module = "com.amazonaws:aws-java-sdk-cloudwatch", version = { ref = "awsSdk1" } }
awsSdk1-core = { module = "com.amazonaws:aws-java-sdk-core", version = { ref = "awsSdk1" } }
awsSdk1-dynamodb = { module = "com.amazonaws:aws-java-sdk-dynamodb", version = { ref = "awsSdk1" } }
awsSdk1-encryption = { module = "com.amazonaws:aws-encryption-sdk-java", version = { ref = "awsEncryptionSdk" } }
awsSdk1-glue = { module = "com.amazonaws:aws-java-sdk-glue", version = { ref = "awsSdk1" } }
```



```
json {
    target '**/*.json'
    targetExclude 'tmp/**/*.json', '**/.terraform/**/*.json',

    toggleOffOn()

    gson()
            .indentWithSpaces(2)
            .version("${libs.versions.gson.get()}")
            .escapeHtml()
}
```

```
exclude module: "slf4j-simple"
exclude module: "slf4j-timbre"
exclude module: "spring-boot-starter-logging"

resolutionStrategy {
    failOnVersionConflict()

    if (!gradle.startParameter.taskRequests.any({ it.args.contains('dependencyUpdates') })) {
        force "ch.qos.logback:logback-classic:${libs.versions.logback.get()}"
        force "ch.qos.logback:logback-core:${libs.versions.logback.get()}"
        force "com.amazonaws:aws-java-sdk-cloudwatch:${libs.versions.awsSdk1.get()}"
        force "com.amazonaws:aws-java-sdk-core:${libs.versions.awsSdk1.get()}"
        force "com.amazonaws:aws-java-sdk-dynamodb:${libs.versions.awsSdk1.get()}"
        force "com.amazonaws:aws-java-sdk-glue:${libs.versions.awsSdk1.get()}"
        force "com.amazonaws:aws-java-sdk-kms:${libs.versions.awsSdk1.get()}"
        force "com.amazonaws:aws-java-sdk-s3:${libs.versions.awsSdk1.get()}"
```



```
protobuf {
    protoc {
        artifact = "com.google.protobuf:protoc:${libs.versions.protobuf.get()}"
    }
}
```

TL;DR

Choice is counter-productive

Don’t rely on engineer’s action

Think a lot about development experience

Be explicit about decisions

If You Liked It, Then You Shoulda Put a *CI Gate* on It

Forcing things explicit helps to avoid mistakes

```
int dependencyBlockIndex = 0;
for (String buildGradleLine : FileUtils.readLines(buildGradleFile, StandardCharsets.UTF_8)) {
    if (StringUtils.containsIgnoreCase(buildGradleLine, "dependencies {")) {
        dependencyBlockIndex++;
        continue;
    } else if (dependencyBlockIndex > 0 && StringUtils.containsIgnoreCase(buildGradleLine, "{")) {
        dependencyBlockIndex++;
    }
    if (dependencyBlockIndex > 0 && StringUtils.containsIgnoreCase(buildGradleLine, "}")) {
        dependencyBlockIndex--;
    }

    String dependencyLine = StringUtils.trim(buildGradleLine);

    var skipLines = new String[] {"artifact {", "classifier = \"osx-aarch_64\"", "}"};
    if (StringUtils.equalsAny(dependencyLine, skipLines)) {
        continue;
    }

    var allowedDependencySources = new String[] {"libs.", "projects.", "localGroovy()",
            "files(\"${System.properties['java.home']}/lib/"};
    if (dependencyBlockIndex > 0 &&
            StringUtils.isNotBlank(dependencyLine) &&
            !StringUtils.containsAny(dependencyLine, allowedDependencySources)) {
        dependenciesChecks.check(allDependenciesDefinedViaCatalogueCheck)
                .failed(
                        "dependency [" + dependencyLine + "] is defined in " +
                                StringUtils.remove(buildGradleFile.getAbsolutePath(), projectPath + "/") +
                                " instead of gradle/libs.versions.toml");
    }

    if (StringUtils.startsWithIgnoreCase(dependencyLine, "force \"")) {
        String dependencyModule = StringUtils.substring(
                dependencyLine,
                "force \"".length(),
                StringUtils.lastIndexOf(dependencyLine, ":"));
        if (dependenciesCatalogue.getLibraries()
                .values()
                .stream()
                .noneMatch(library -> StringUtils.equals(library.getModule(), dependencyModule))) {
            dependenciesChecks.check(allDependenciesDefinedViaCatalogueCheck)
                    .failed(
                            "dependency [" + dependencyLine + "] is present in " +
```

TL;DR

Choice is counter-productive

Don’t rely on engineer’s action

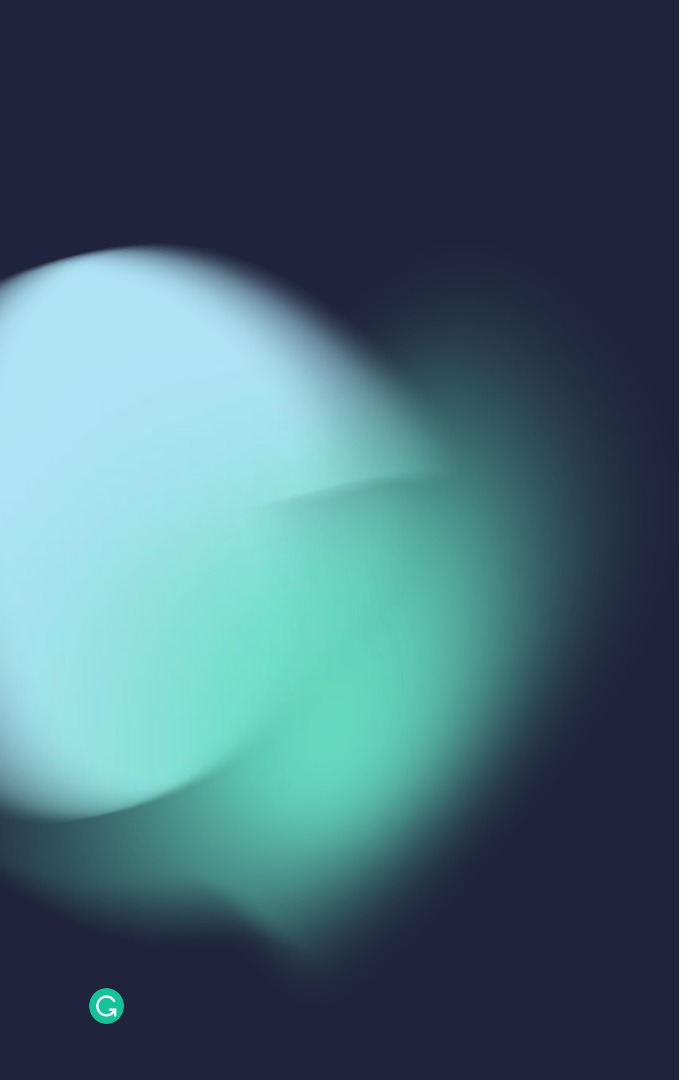
Think a lot about development experience

Be explicit about decisions

If You Liked It, Then You Shoulda Put a *CI Gate* on It

Forcing things explicit helps to avoid mistakes

The ugliest ad-hoc script is better than the prettiest documentation



```
181 actionable tasks: 83 executed, 98 from cache
$ ./gradlew dependency-analysis:checkDependencies --refresh-dependencies --no-daemon
To honour the JVM settings for this build a single-use Daemon process will be forked. See https://
emon.
Daemon will be stopped at the end of the build
Type-safe project accessors is an incubating feature.
> Task :build-tools:compileJava FROM-CACHE
> Task :build-tools:processResources NO-SOURCE
> Task :build-tools:classes UP-TO-DATE
> Task :build-tools:jar
> Task :dependency-analysis:compileJava FROM-CACHE
> Task :dependency-analysis:processResources NO-SOURCE
> Task :dependency-analysis:classes UP-TO-DATE
> Task :dependency-analysis:checkDependencies
build.gradle dependencies defined through catalogue: LOOKS GOOD
libs.versions.toml versions: LOOKS GOOD
renovate configuration: LOOKS GOOD
libs.versions.toml libraries: LOOKS GOOD
libs.versions.toml bundles: LOOKS GOOD
libs.versions.toml plugins: LOOKS GOOD
Dependencies check succeeded
BUILD SUCCESSFUL in 10s
```

TL;DR

Choice is counter-productive

Don’t rely on engineer’s action

Think a lot about development experience

Be explicit about decisions

If You Liked It, Then You Shoulda Put a *CI Gate* on It

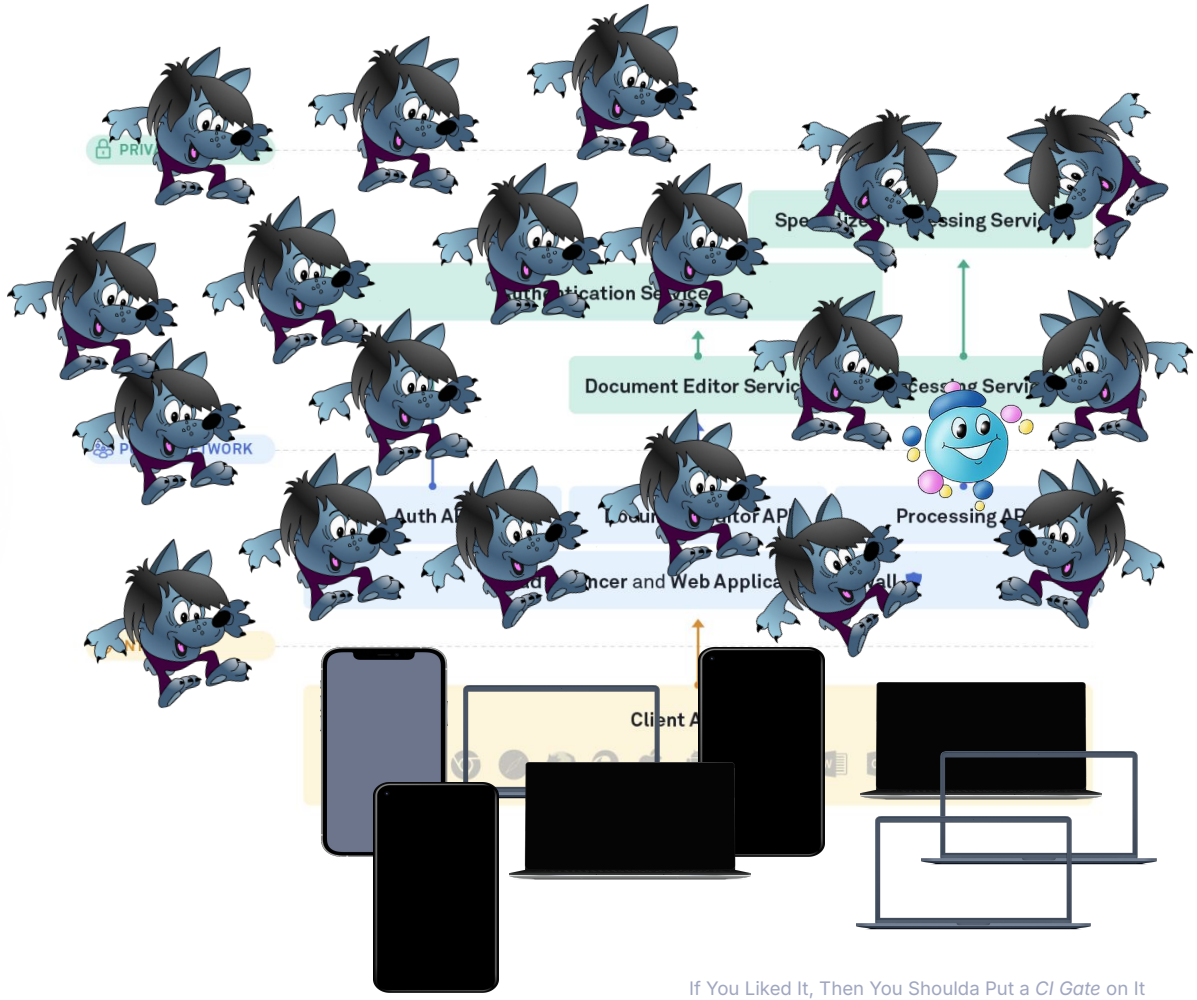
Forcing things explicit helps to avoid mistakes

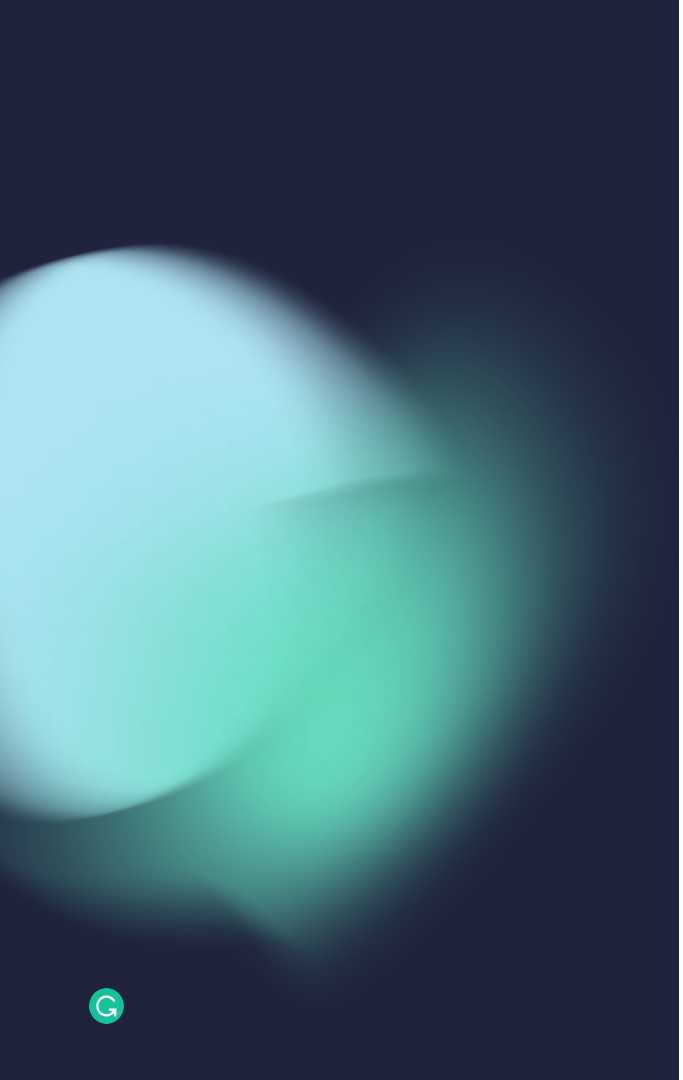
The ugliest ad-hoc script is better than the prettiest documentation

Join us!

Software Engineer,
Core Data

Bonus





Revapi Site / Revapi

Revapi

Full featured API checker for Java and beyond.

Revapi is an API analysis and change tracking tool written in Java.

Its focus is mainly on Java language itself but it has been specifically designed to not be limited to just Java. API is **much more** than just java classes - also various configuration files, schemas, etc. can contribute to it and users can become reliant on them.

NOTE

Revapi is in beta. The Java API checker is fairly capable and can track both binary and source compatibility and the maven plugin and ant task are fairly useful but there are still many things to be done and polished. Your help is greatly appreciated.

Why?

Because surprisingly there doesn't seem to exist a simple yet extensible, developer-oriented tool that could be used to check the APIs and, more importantly, track their evolution. APIs are not static, they evolve to accommodate new features and past mistakes but at the same time, each change in the APIs potentially breaks the code of the users of the APIs.

It is therefore important for the tool to do 2 things right:

```
→  common-api git:(y-fun-1) ✗ ./gradlew revapi
Type-safe project accessors is an incubating feature.
> Task :capi-lib:revapi FAILED

FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':capi-lib:revapi'.
> There were Java public API/ABI breaks reported by revapi:

  java.method.numberOfParametersChanged: The number of parameters of the method have changed.

  old: method [illegible]::getCard(java.lang.String)
  new: method [illegible]::getCard(java.lang.String, java.lang.String)

  SOURCE: BREAKING, BINARY: BREAKING
```

Revapi Site / Revapi

Revapi

Full featured API checker for Java and beyond.

[illegible] could be used to check the APIs and, more importantly, track their evolution. APIs are not static, they evolve to accommodate new features and past mistakes but at the same time, each change in the APIs potentially breaks the code of the users of the APIs.

It is therefore important for the tool to do 2 things right:

japicmp

japicmp is a tool to compare two versions of a jar archive:

```
java -jar japicmp-0.17.2-jar-with-dependencies.jar -n new-version.jar -o old-version.jar
```

It can also be used as a library:

```
JarArchiveComparatorOptions comparatorOptions = new JarArchiveComparatorOptions();
JarArchiveComparator jarArchiveComparator = new JarArchiveComparator(comparatorOptions);
List<JApiClass> jApiClasses = jarArchiveComparator.compare(oldArchives, newArchives);
```

japicmp is available in the Maven Central Repository:

```
<dependency>
        <groupId>com.github.siom79.japicmp</groupId>
        <artifactId>japicmp</artifactId>
        <version>0.17.2</version>
</dependency>
```

A maven plugin allows you to integrate the checks into your build:

```
<plugin>
        <groupId>com.github.siom79.japicmp</groupId>
        <artifactId>japicmp-maven-plugin</artifactId>
        <version>0.17.2</version>
        <configuration>
                <oldVersion>
                        <dependency>
                                <groupId>japicmp</groupId>
                                <artifactId>japicmp-test-v1</artifactId>
                                <version>${oldversion}</version>
                                <type>jar</type>
                        </dependency>
                </oldVersion>
```

japicmp

japicmp is a tool to compare two versions of a jar archive:

```
java -jar japicmp-0.17.2-jar-with-dependencies.jar -n new-version.jar -o old-version.jar
```

It can also be used as a library:

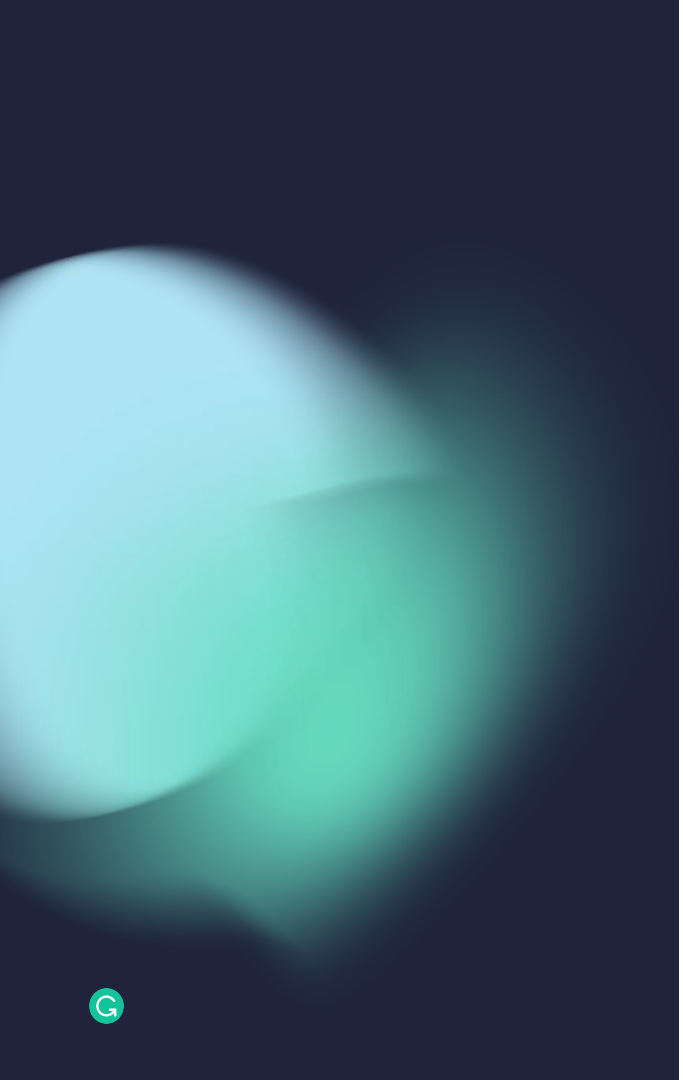
```
JarArchiveComparatorOptions comparatorOptions = new JarArchiveComparatorOptions();
JarArchiveComparator jarArchiveComparator = new JarArchiveComparator(comparatorOptions);
List<JApiClass> jApiClasses = jarArchiveComparator.compare(oldArchives, newArchives);
```

japicmp is available in the Maven Central Repository:

```
<groupId>com.github.siom79.japicmp</groupId>
<artifactId>japicmp-maven-plugin</artifactId>
<version>0.17.2</version>
<configuration>
        <oldVersion>
                <dependency>
                        <groupId>japicmp</groupId>
                        <artifactId>japicmp-test-v1</artifactId>
                        <version>${oldversion}</version>
                        <type>jar</type>
                </dependency>
        </oldVersion>
```

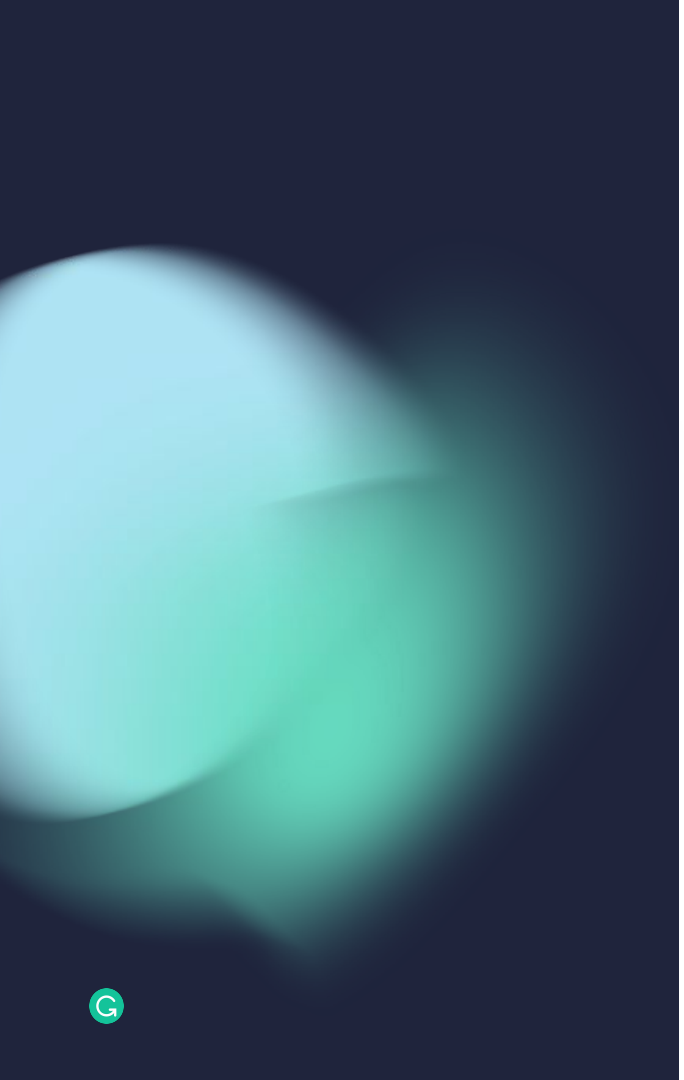
```
cat: capi-lib/reports/japi.txt: No such file or directory
➜  common-api git:(y-fun-1) ✗ cat capi-lib/build/reports/japi.txt
Comparing source compatibility of  against
***! MODIFIED CLASS: PUBLIC grammarly.capi.lib.cards.MistakeCards  (not serializable)
       ===  CLASS FILE FORMAT VERSION: 58.0 <- 58.0
       ---! REMOVED METHOD: PUBLIC(-) STATIC(-) grammarly.capi.lib.api.sprawl.MistakeCard getCard(java.lang.String)
       +++  NEW METHOD: PUBLIC(+) STATIC(+) grammarly.capi.lib.api.sprawl.MistakeCard getCard(java.lang.String, java.lang.String)
```

```
tasks.register('japicmp', me.champeau.gradle.japicmp.JapicmpTask) {
    oldClasspath.from(files('capi-lib-0.11.3.jar'))
    newClasspath.from(tasks.named('jar'))

    onlyModified = true
    failOnModification = true

    txtOutputFile = layout.buildDirectory.file('reports/japi.txt')
}
```



```
    baseline "${project.group}:${project.moduleName}:+"
}

def baselineConf = configurations.create('japicmp-baseline') {
    extendsFrom configurations.baseline
}

tasks.register('japicmp', me.champeau.gradle.japicmp.JapicmpTask) {
    oldClasspath.from(baselineConf.resolve().findAll { it.name.contains project.moduleName })
    newClasspath.from(tasks.named('jar'))

    onlyModified = false

    failOnModification = false
    failOnSourceIncompatibility = false

    txtOutputFile = layout.buildDirectory.file('reports/japi/japi.txt')
}
```

```
    }

    publications { PublicationContainer it ->
        maven(MavenPublication) {
            artifactId = project.moduleName
            version = new XmlSlurper()
                    .parse(new File(project.buildDir, 'reports/japi/japi.xml'))
                        .'classes'.'*'.'@binaryCompatible'.every({ it == 'true' })
                            ? "COMPATIBLE"
                            : "NON_COMPATIBLE"

            from components.java
        }
    }
}
```



```
$ export GRADLE_USER_HOME=.gradle
$ cat >> ./gradle.properties <<EOF # collapsed multi-line command
$ ./gradlew capi-lib:japicmp --refresh-dependencies --stacktrace --info --no-daemon
Downloading https://services.gradle.org/distributions/gradle-8.1.1-bin.zip
..........10%.....20%.....30%.....40%.....50%.....60%.....
.......100%
Initialized native services in: /builds/dZcwKbp4/1/core/common-api/.gradle/native
Initialized jansi services in: /builds/dZcwKbp4/1/core/common-api/.gradle/native
Welcome to Gradle 8.1.1!
Here are the highlights of this release:
 - Stable configuration cache
 - Experimental Kotlin DSL assignment syntax
 - Building with Java 20
For more details see https://docs.gradle.org/8.1.1/release-notes.html
Received JVM installation metadata from '/opt/jdk-20': {JAVA_HOME=/opt/jdk-20, JAVA_VERSION=20, JAVA_
RUNTIME_VERSION=20+36-2344, VM_NAME=OpenJDK 64-Bit Server VM, VM_VERSION=20+36-2344, VM_VENDOR=Oracle
Checking if the launcher JVM can be re-used for build. To be re-used, the launcher JVM needs to match
=512m -XX:+HeapDumpOnOutOfMemoryError --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.bas
NNAMED --add-opens=java.prefs/java.util.prefs=ALL-UNNAMED --add-opens=java.base/java.nio.charset=ALL-U
ase/java.util.concurrent.atomic=ALL-UNNAMED -Xmx2g -Dfile.encoding=UTF-8 -Duser.country -Duser.languag
To honour the JVM settings for this build a single-use Daemon process will be forked. See https://docs
emon.
Starting process 'Gradle build daemon'. Working directory: /builds/dZcwKbp4/1/core/common-api/.gradle/
m -XX:+HeapDumpOnOutOfMemoryError --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/ja
ED --add-opens=java.prefs/java.util.prefs=ALL-UNNAMED --add-opens=java.base/java.nio.charset=ALL-UNNAM
java.util.concurrent.atomic=ALL-UNNAMED -Xmx2g -Dfile.encoding=UTF-8 -Duser.country -Duser.language=en
er/dists/gradle-8.1.1-bin/9wiye5v2saajue4irfo8ybqfp/gradle-8.1.1/lib/gradle-launcher-8.1.1.jar -javaag
8.1.1-bin/9wiye5v2saajue4irfo8ybqfp/gradle-8.1.1/lib/agents/gradle-instrumentation-agent-8.1.1.jar org
Successfully started process 'Gradle build daemon'
An attempt to start the daemon took 0.551 secs.
```

TL;DR

Choice is counter-productive

Don’t rely on engineer’s action

Think a lot about development experience

Be explicit about decisions

If You Liked It, Then You Shoulda Put a *CI Gate* on It

Forcing things explicit helps to avoid mistakes

The ugliest ad-hoc script is better than the prettiest documentation