


UNPLEASANT WORDS AND HOW TO DETEKT THEM

JAROSŁAW MICHALIK
Android Tech Talks #24


LTE 11:15

← Odbiorca zamówienia

☒ Tak, akceptuję **Regulamin** serwisu.

☒ Tak, chcę być informowany o akcjach marketingowych i ofertach specjalnych.

☒ Wyrażam zgodę na przekazanie moich danych osobowych spółkom z grupy Am Rest.

* pole obowiązkowe

ZAPISZ

1 2 3 4 5 6 7 8 9 0
 q w e r t y u i o p
 a s d f g h j k l
 ↑ z x c v b n m ×
 ? 1 @ . →


Krzysztof ZoZo Morzyc ▶ **Pizza Hut**

23 hrs · 🌐

Hej @pizzahutpolska Doceniam humor programisty 😊 ale chyba trzeba to zmienić.

👍 Like 💬 Comment ➦ Share

👍 😂 ❤️ 113 Top Comments ▾

18 shares 5 Comments


Pizza Hut 🟢 😊 😊 ups, dzięki za info, już sprawdzamy o co chodzi!

Like · Reply · 👍 16 · 23 hrs


Petter Van Rijn tylko nie róbcie krzywdy programiście... ma chłopak poczucie humoru. Krótkie, zwarte i żołnierskie, ale ma 😊

Like · Reply · 👍 21 · 23 hrs


Wiktor Harężlak xD

Like · Reply · 20 hrs


Jacek JJack Zawistowski nie ma co sprawdzać o co chodzi 😊 Komunikat świetnie to tłumaczy 😊

Like · Reply · 👍 1 · 17 hrs


Bartosz Szczepiński Przecież widać. Zjechało się.

Like · Reply · 👍 7 · 14 hrs

Write a reply...

Write a comment...

HOW DID THAT HAPPEN?

- Frustration?
- Temporary / prototyping values - proper text was not ready yet
- Debugging without proper debugger

WHY DID THAT HAPPEN?

- Someone decided to write that...
- Lack of proper code review
- Not enough testing or no testing at all!

WHEN YOU TRUST REFACTORING TOO MUCH...

```
class PubDisplayableDupaMapper: Mapper<PubModel, PubDisplayable>
```

```
class PubDisplayableMapper: Mapper<PubModel, PubDisplayable>
```

```
class ViewModel @Inject constructor(  
    val pubDisplayableDupaMapper: Mapper<PubModel, PubDisplayable>  
) {
```

PREVENTION IS BETTER THAN
CURE

THE SIMPLEST SOLUTION

```
grep -nr 'badword' .
```

It gets the work done, but eventually you will end up building custom lint.

KOTLIN STATIC CODE ANALYSIS

- Android Lint
- Ktlint
- Detekt

<https://github.com/vanniktech/kotlin-on-code-quality-tools>

DETEKT

- Code smell analysis for Kotlin projects
- Generates complexity reports
- Configurable & extensible
- <https://github.com/arturbosch/detekt>

DETEKT

- Gradle plugin
- Has IntelliJ plugin!
- Works fine with `@Suppress` annotations
- Can be used in non-android projects

WE ALREADY HAD DETEKT IN PROJECT... SO



Mobile Apps / MA-860

**Create custom detekt rule for
prototyping/debugging artifacts**

NEW GRADLE MODULE

```
apply plugin: "kotlin"

repositories {
    jcenter()
}

dependencies {
    implementation common.kotlinDependencies.values()
    implementation "io.gitlab.arturbosch.detekt:detekt-api:$detektVersion"
    implementation "io.gitlab.arturbosch.detekt:detekt-core:$detektVersion"

    implementation „org.jetbrains.kotlin:kotlin-compiler-embeddable"$kotlinVersion"

    implementation unitTest["junit5"].values()
    testRuntimeOnly unitTest["junit5Engines"].values()

    testImplementation "io.gitlab.arturbosch.detekt:detekt-api:$detektVersion"
    testImplementation "io.gitlab.arturbosch.detekt:detekt-test:$detektVersion"
}
```

```
#build.gradle
```

```
dependencies {  
    detektPlugins project(":detekt-rule")  
}
```

```
class BadWordRuleSetProvider : RuleSetProvider {  
    override val ruleSetId: String = "badword"  
  
    override fun instance(config: Config): RuleSet {  
        return RuleSet(ruleSetId, listOf(BadWordRule(config)))  
    }  
}
```

REPORTING AN ISSUE

```
/**
 * An issue represents a problem in the codebase.
 */
data class Issue(val id: String,
                 val severity: Severity,
                 val description: String,
                 val debt: Debt)

/**
 * Rules can be classified into different severity grades.
 * Maintainer can choose
 * a grade which is most harmful to their projects.
 */
enum class Severity {
    CodeSmell, Style,
    Warning, Defect, Minor,
    Maintainability, Security, Performance
}

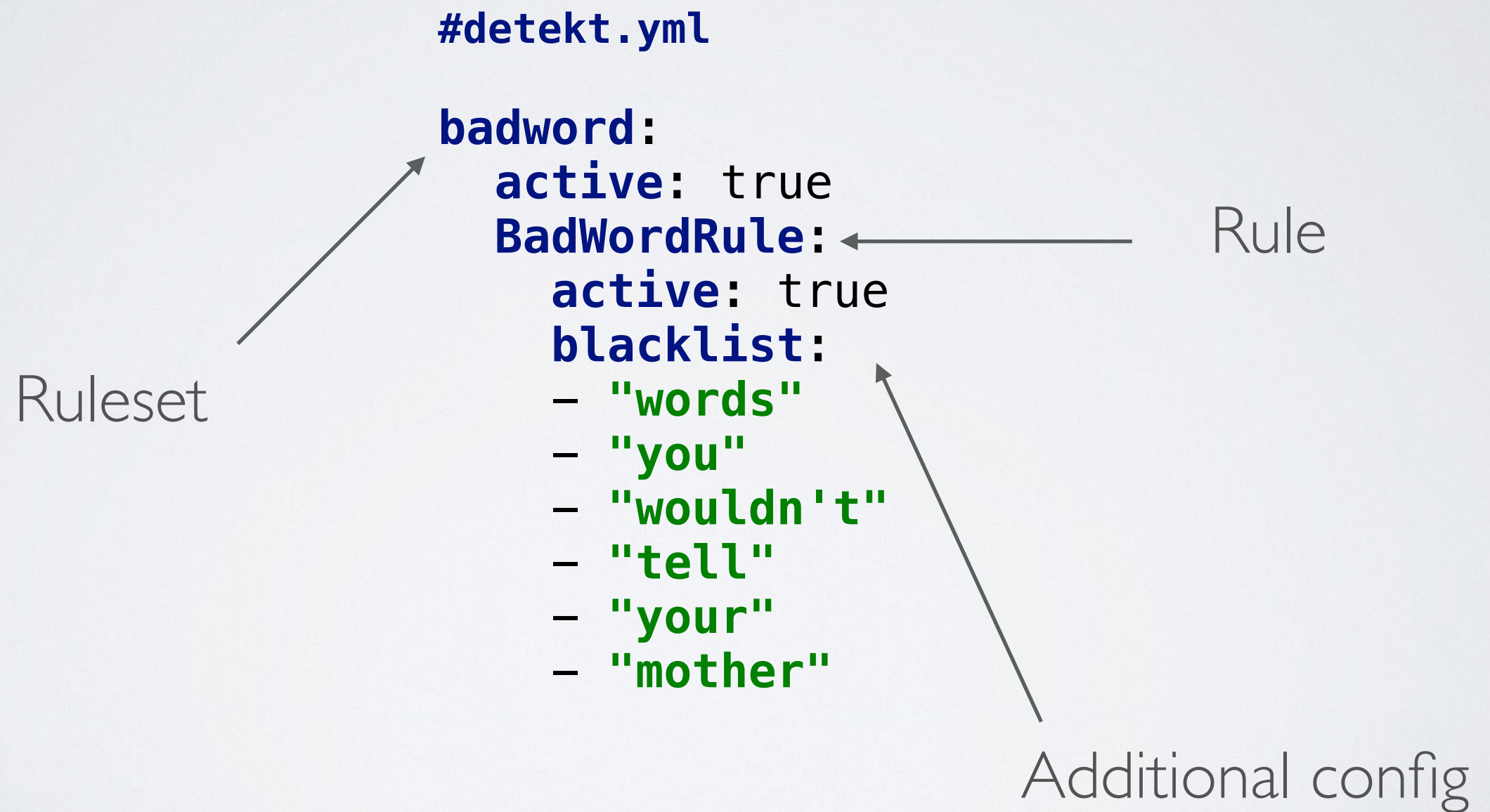
/**
 * Debt describes the estimated amount of work needed to fix a given issue.
 */
data class Debt(val days: Int = 0, val hours: Int = 0, val mins: Int = 0)
```



```
class BadWordRule(config: Config = Config.empty) : Rule(config) {  
    override val issue: Issue = Issue(  
        id = javaClass.simpleName,  
        severity = Severity.Style,  
        description = "Find bad words in project files.",  
        debt = Debt.FIVE_MINS)  
  
    val blacklist by lazy {  
        config.subConfig(„BadWordRule")  
            .valueOrDefault("blacklist", emptyList<String>())  
    }  
  
    override fun visit(root: KtFile) {  
        validate(root)  
    }  
  
    private fun validate(element: PsiElement) {  
        blacklist.forEach {  
            if (element.text.toString().contains(it.toLowerCase())) {  
                report(CodeSmell(  
                    issue = issue,  
                    message = "Forbidden word: $it",  
                    entity = Entity.from(element)  
                ))  
            }  
        }  
    }  
}
```

```
override fun visitAnnotatedExpression(expression: KtAnnotatedExpression) {  
    super.visitAnnotatedExpression(expression)  
}  
  
override fun visitAnnotation(annotation: KtAnnotation) {  
    super.visitAnnotation(annotation)  
}  
  
override fun visitClass(klass: KtClass) {  
    super.visitClass(klass)  
}  
  
override fun visitBinaryFile(file: PsiBinaryFile?) {  
    super.visitBinaryFile(file)  
}  
  
override fun visitCallExpression(expression: KtCallExpression) {  
    super.visitCallExpression(expression)  
}  
  
override fun visitNamedDeclaration(declaration: KtNamedDeclaration) {  
    super.visitNamedDeclaration(declaration)  
}
```

CONFIGURATION



REPORTS

```
./gradlew app:detekt
```

```
Ruleset: badword – 10min debt
```

```
BadWordRule – [App.kt] at MyApp/app/src/main/java/com/myapp/App.kt:1:1
```

```
BadWordRule – [Repository.kt] at MyApp/app/src/main/java/com/myapp/market/  
Repository.kt:1:1
```

BONUS!

- Detekt checks your .kts file too!
- You can enable auto-formatter (based on Ktlint)

USE IN CONTINUOUS INTEGRATION

Detekt plugs itself in into

- **./gradlew check**

task automatically

But it's better to use

- **./gradlew detekt**

as separate CI step

USE GIT HOOKS

```
cd YOURREPO/.git/hooks
```

```
ls -l
```

```
applypatch-msg.sample  
fsmonitor-watchman.sample  
pre-applypatch.sample  
pre-commit.sample  
pre-rebase.sample  
prepare-commit-msg.sample  
commit-msg.sample  
post-update.sample  
pre-commit  
pre-push.sample  
pre-receive.sample  
update.sample
```

ADD GIT HOOKS

Run your checks before
commit or before push

```
# pre-commit
```

```
./gradlew detekt
```

WHAT IS INAPPROPRIATE?

- Swearword in one language but not in other
- Some medical terms are used as swears
- Vulgarness depends on context



Change the string "allopenissues" to not include the word "penis"

▼ Details

Type:  Suggestion

Status: **RESOLVED** ([View Workflow](#))

Affects Version/s: None


Resolution: Unsolved Mysteries

Component/s: None

Fix Version/s: None

Labels: None

Feedback Policy:

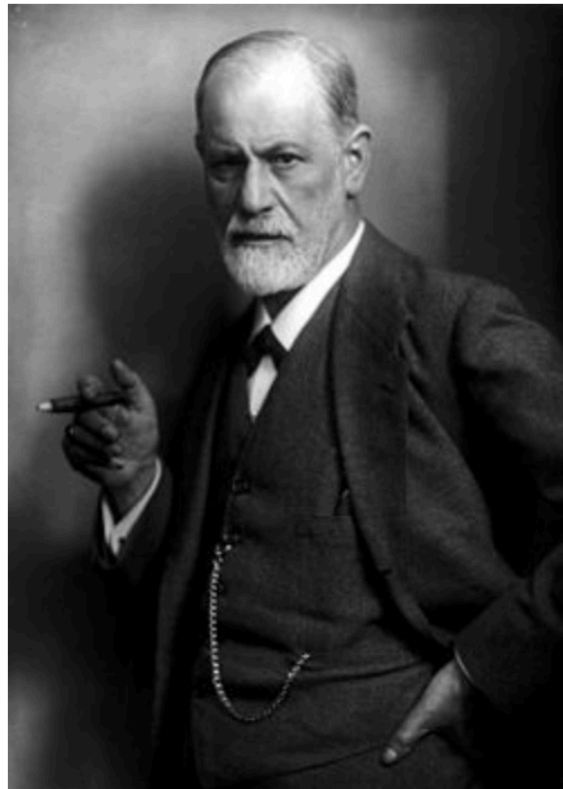
 We collect Jira feedback from various sources, and we evaluate what we've collected when planning our product roadmap. To understand how this piece of feedback will be reviewed, see [An updated workflow for server feature suggestions](#).

Current Status:



Atlassian Update – 24-01-2019

Sigmund Freud: Sometimes a Cigar is Just a Cigar.



Regards,
Jira Server Team



LTE



11:15



Odbiorca zamówienia



Tak, akceptuję Regulamin serwisu.



Tak, chcę być informowany o akcjach marketingowych i ofertach specjalnych.



Wyrażam zgodę na przekazanie moich danych osobowych spółkom z grupy AmRest



* pole obowiązkowe



Naprawiło się!

ZAPISZ

1

2

3

4

5

6

7

8

9

0

q

w

e

r

t

y

u

i

o

p

a

s

d

f

g

h

j

k

l



z

x

c

v

b

n

m



?1☺

@



CONTACT

- <https://github.com/rozkminiacz>
- <https://medium.com/@rozkminiacz>
- <https://twitter.com/rozkminia>